

Arm® Keil® MDK Version 5 for SimpleLink™ MSP432™ Microcontrollers

This user's guide describes the use of the Arm® Keil® MDK version 5 with the SimpleLink™ MSP432™ low-power microcontrollers (MCUs).

Contents

1	Installing Keil µVision 5 IDE	4
2	Installing Device Family Packs.....	4
2.1	Installing Using a Downloaded SimpleLink MSP432P4xx Device Family Pack	4
2.2	Installing Using the Keil Pack Installer	6
3	Creating a SimpleLink MSP432 MCU Project From Scratch	7
4	Opening an Existing SimpleLink MSP432 MCU Project	14
5	Using SimpleLink MSP432 MCU Examples From the Software Development Kit	15
6	Debugging the Application	16
6.1	Using Segger J-Link Debug Probe.....	17
6.2	Using Keil ULINK2 and ULINK Pro debug probe	25
6.3	Using XDS110-ET Debug Probe	29
6.4	Debugging Driver Lib in ROM	33
6.5	Using Serial Wire Output (SWO) Hardware Trace Analyzer	35
6.6	Using ETM Trace (MSP432E4 Devices Only).....	37
7	Frequently Asked Questions.....	41
8	Additional Keil MDK-Arm Information.....	42
9	References	42

List of Figures

1	Keil Pack Unzip Tool	4
2	License Agreement.....	5
3	Device Family Pack Has Been Installed Successfully	5
4	Launching the Pack Installer	6
5	Installing a MSP432 Device Family Pack	7
6	Create New Project.....	7
7	Assign Project Name	8
8	Select MSP432P401R Device	9
9	Add Run-Time Environment Components to the Project	10
10	Empty Project.....	11
11	Add New Item to Project	11
12	Add User Code Template	12
13	Compiling the Project.....	13
14	Opening a Project.....	14
15	Choosing a Project to Import	15
16	Select Project From SDK.....	16
17	Project Options	17
18	Select J-Link Debug Probe	17
19	Go to J-Link Debug Probe Settings Window	18

20	Enable J-Link Download Options	19
21	J-Link Download Function Configuration.....	20
22	Start Debug Session	20
23	Messages During Debug Session Startup	21
24	Debug Session Halted at main()	22
25	Using J-Link Commander to Enable Power Output to Target System	22
26	Launch J-Link Control Panel.....	23
27	J-Link Control Panel	24
28	J-Link Script Detecting That the Device Has Been Secured	24
29	Device Has Been Unlocked and Erased	24
30	Project Options	25
31	Selecting the ULINK Pro Debug Probe	25
32	Selecting ULINK Pro Download Options	26
33	Select ULINK Pro Download Function	26
34	Start Debug Session	27
35	Debug Session Successfully Started and Halted at main().....	28
36	Dialog Box Asking to Perform a Factory Reset	28
37	Project Options	29
38	Selecting the XDS110 Debug Probe	30
39	Selecting CMSIS-DAP Settings	30
40	Select ULINK Pro Download Function	31
41	Start Debug Session	31
42	Debug Session Successfully Started and Halted at main().....	32
43	Dialog Box Asking to Perform a Factory Reset	32
44	Adding Path to the Driver Library Folder	33
45	Adding Driver Library to a Project.....	34
46	Adding Initialization File for Debugging	34
47	Stepping Through ROM Driver Lib Source Code	35
48	Debug Settings	35
49	Enabling Trace in Project Options	36
50	Exception Trace	37
51	ETM Debug Probe Setup	38
52	Adding Trace Initialization File.....	39
53	Show Data Trace Window	40
54	Changing Debugger Setting to SWD	41

Trademarks

SimpleLink, MSP432, E2E, LaunchPad are trademarks of Texas Instruments.
 Arm, Keil, µVision, CoreSight are registered trademarks of Arm Limited.
 All other trademarks are the property of their respective owners.

Preface: Read This First

How to Use This User's Guide

This user's guide describes only the features of the Keil MDK 5 that are specific to the MSP432 low-power microcontrollers. It does not fully describe the MSP432 microcontrollers or the complete development software and hardware systems. For details on these items, see the appropriate TI documents listed in [Important MSP432 MCU Documents on the Web](#).

Important MSP432 MCU Documents on the Web

The primary sources of information on the MSP432 MCU family are the device-specific data sheets and user's guides. The [MSP432 website](#) contains the most recent version of these documents.

Documents that describe the Keil MDK 5 IDE can be found on [www.keil.com](#). The Texas Instruments E2E™ Community support forums at [e2e.ti.com](#) can provide additional help.

Documentation for third-party tools, such as the Keil ULINK2 and ULINK Pro debug probes or the Segger J-Link debug probe, can usually be found on the respective third-party website.

If You Need Assistance

Support for the MSP432 microcontrollers and the hardware development tools is provided by the Texas Instruments Product Information Center (PIC). Contact information for the PIC can be found on the TI website at [www.ti.com/support](#). The [TI E2E Community](#) support forums for the MSP432 MCUs provide open interaction with peer engineers, TI engineers, and other experts. Additional device-specific information can be found on the [MSP432 website](#).

1 Installing Keil μ Vision 5 IDE

The Keil μ Vision® 5 IDE can be obtained from the [Keil website](#). MSP432 low-power microcontrollers are supported by Keil μ Vision 5 and higher versions. Previous versions do not support MSP432 MCUs and might not be compatible with MSP432 Device Family Packs.

2 Installing Device Family Packs

The Keil μ Vision 5 IDE does not come with preinstalled device support. To add support for MSP432 MCUs to the IDE, either download an MSP432 Device Family Pack from the TI website (see [Section 2.1](#)), or install the pack using the Keil Pack Installer (see [Section 2.2](#)).

2.1 Installing Using a Downloaded SimpleLink MSP432P4xx Device Family Pack

To download this pack, visit the [MSP432 Device Family Pack page](#) and download the file named TexasInstruments.MSP432P4xx_DFP.<version>.pack, where <version> is the revision of the device family pack. When downloaded completes, double click on the file. The Keil Pack Unzip tool starts and guides you through the installation process.

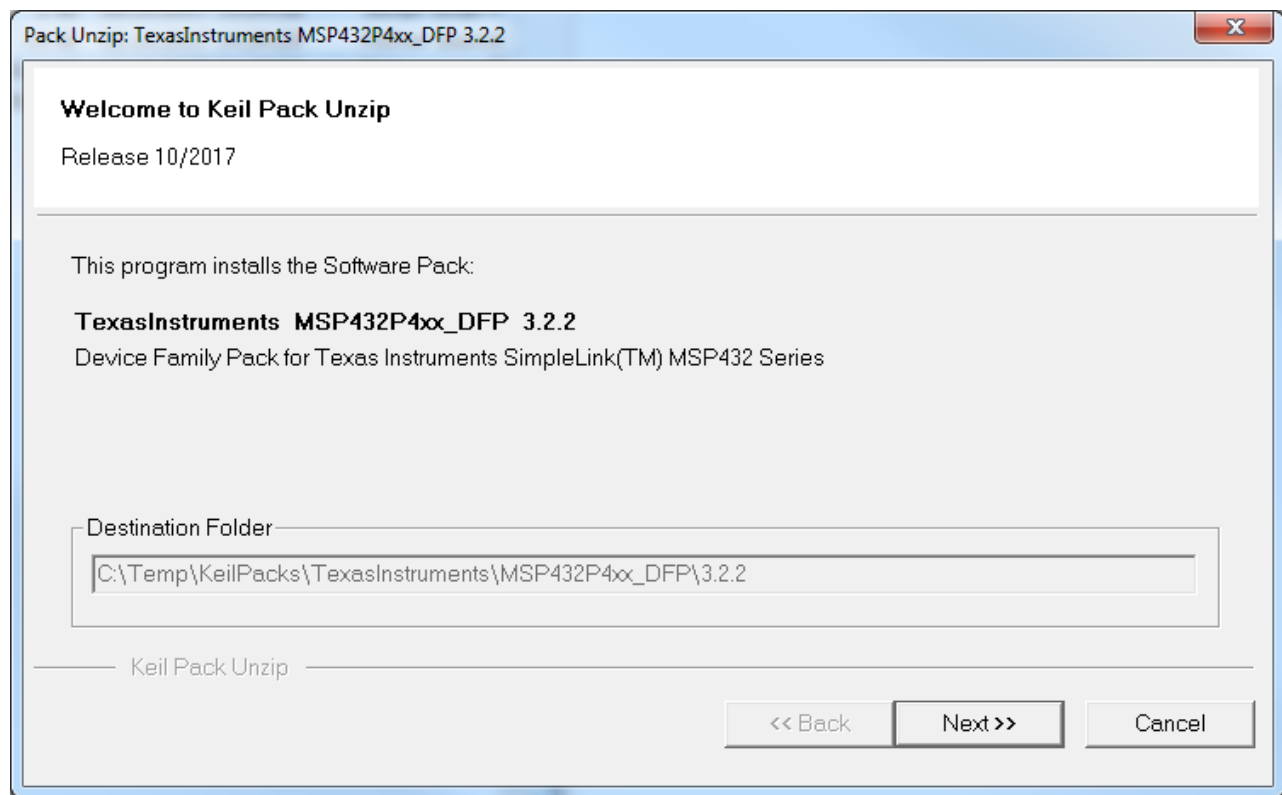


Figure 1. Keil Pack Unzip Tool

Click **Next** to display the TI License Agreement (see [Figure 2](#)).

Take the time to read the agreement before accepting it, then click **Next**.

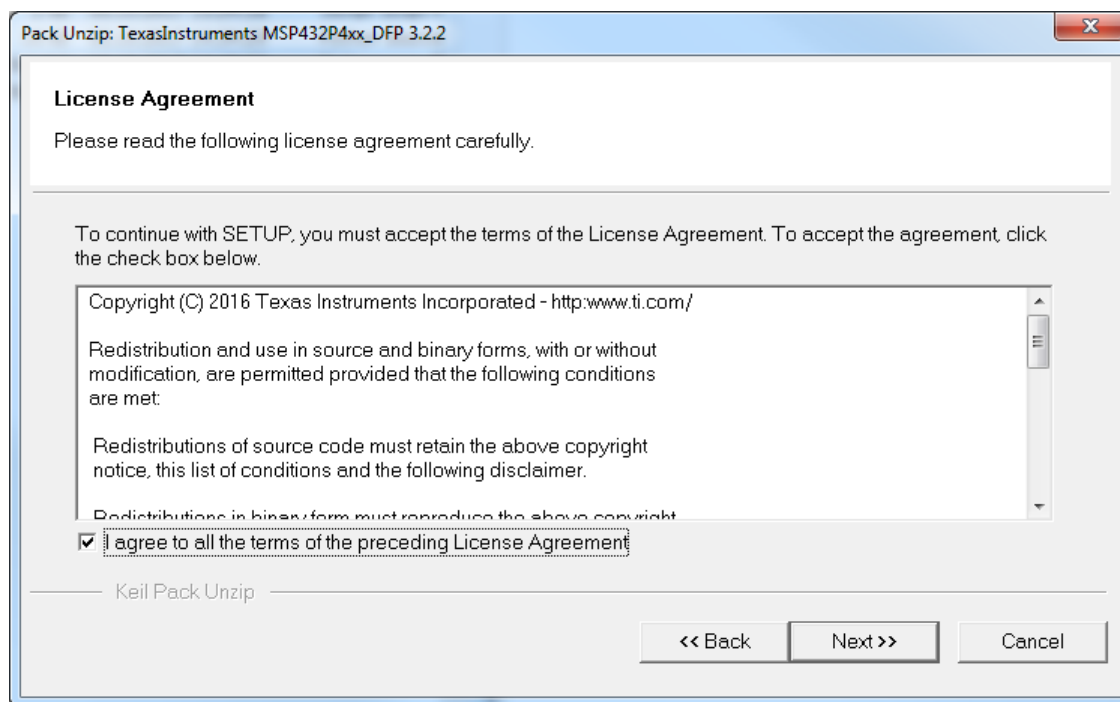


Figure 2. License Agreement

After the Pack Unzip tool has copied the device family files into the μ Vision IDE installation directory, the Pack Unzip tool finishes (see [Figure 3](#)).

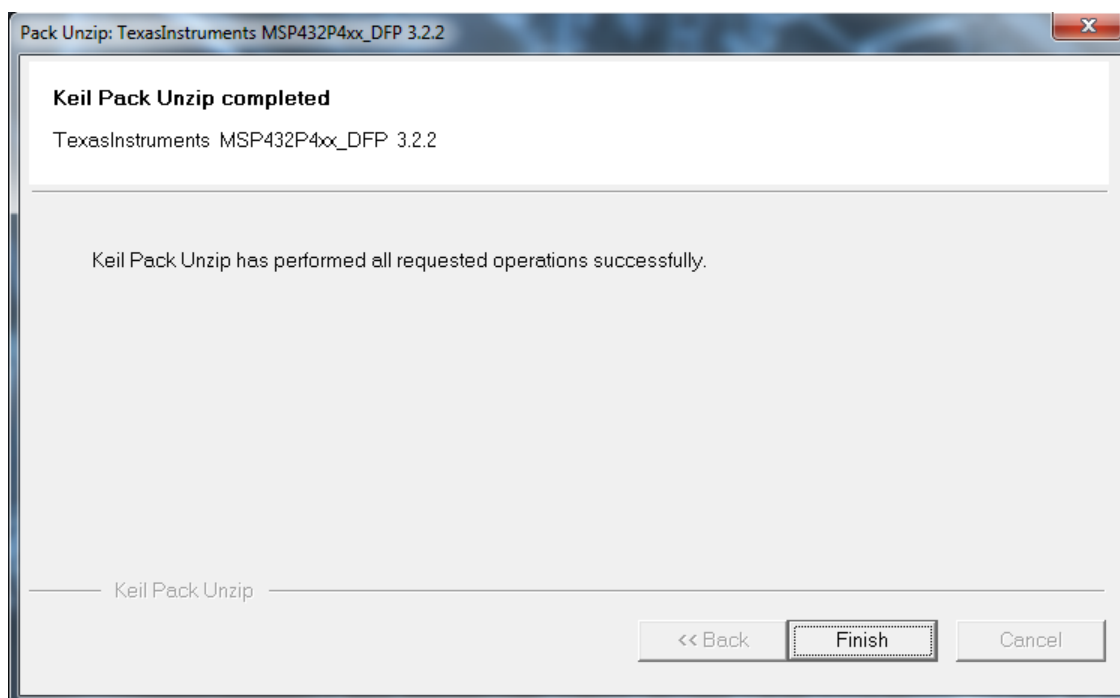


Figure 3. Device Family Pack Has Been Installed Successfully

If the tool reports an error, check whether or not the Device Family Pack has been downloaded correctly, and whether or not the μ Vision IDE directory is writeable.

2.2 Installing Using the Keil Pack Installer

To install using the Keil Pack Installer, open the Keil μ Vision IDE. To launch the Keil Pack Installer, go to **Project** → **Manage** → **Pack Installer** (see Figure 4).

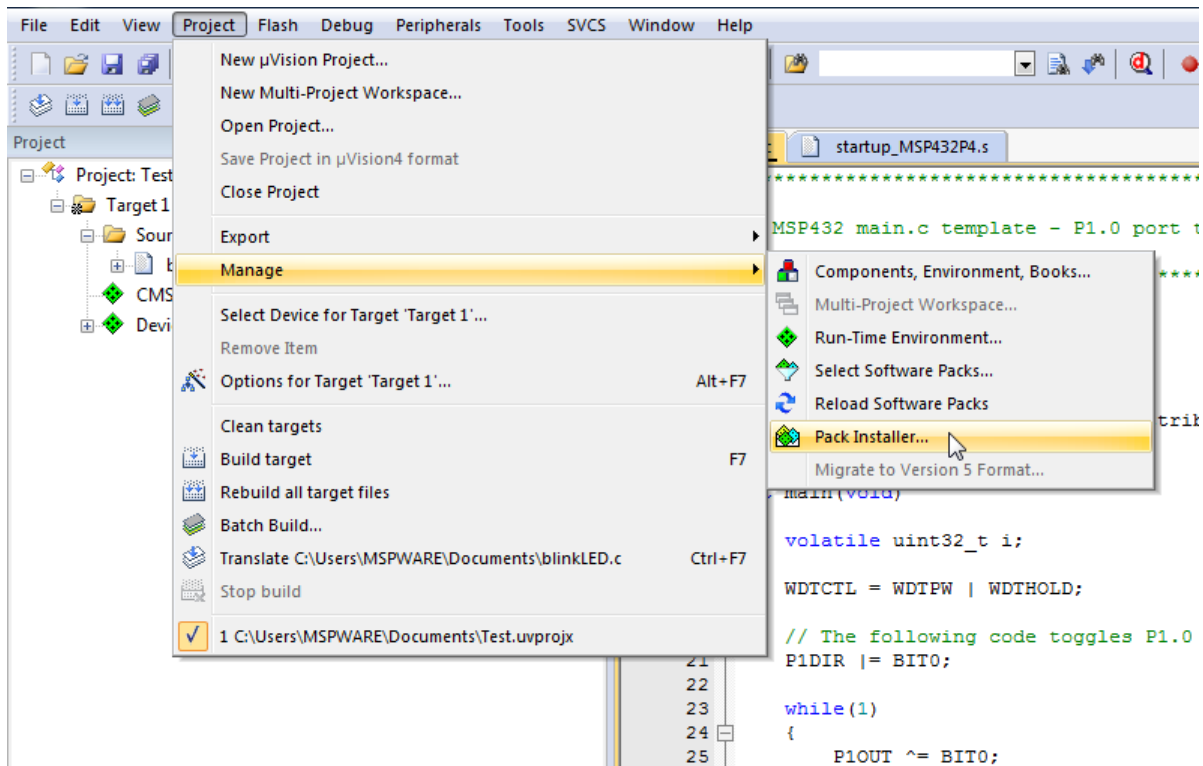


Figure 4. Launching the Pack Installer

When the pack installer has been launched, locate Texas Instruments::MSP432P4xx Series in the devices listing and select the newest version of the associated Texas Instruments::MSP432P4xx_DFP pack (see Figure 5).

Click **Install** to let the IDE download and install the MSP432P4xx Device Family Pack into the IDE.

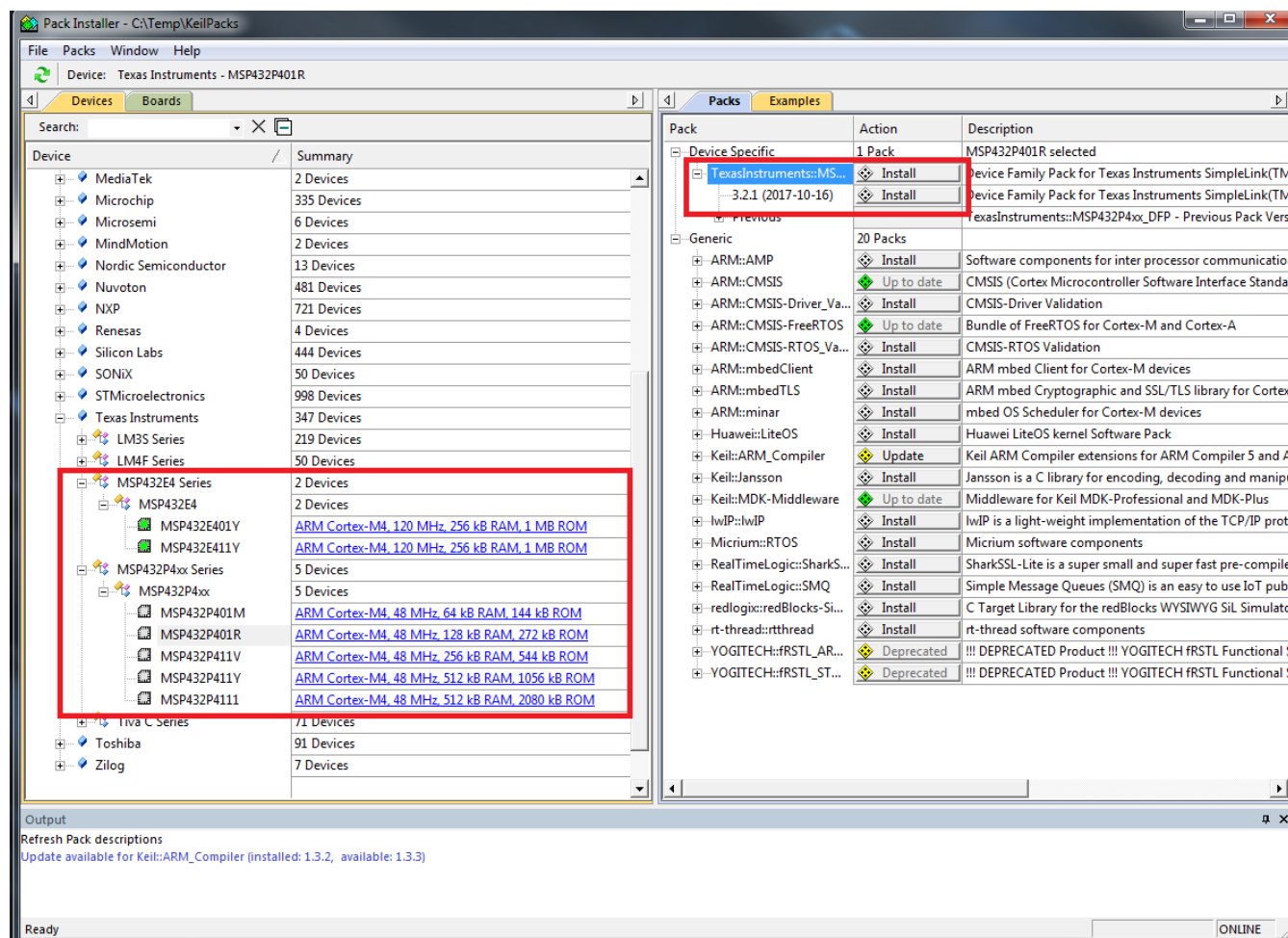


Figure 5. Installing a MSP432 Device Family Pack

If the tool reports an error, check whether the μ Vision IDE directory is writeable.

3 Creating a SimpleLink MSP432 MCU Project From Scratch

When starting μ Vision for the first time, create a project for the MSP432 MCU. Go to **Project** \rightarrow **New μ Vision Project** and click on the selection (see Figure 6).

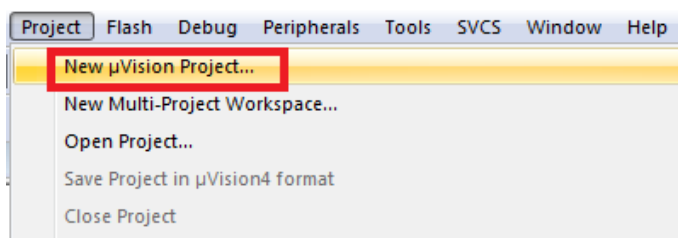


Figure 6. Create New Project

A file dialog (see Figure 7) is displayed to prompt for the project name and location. Navigate to an appropriate directory on the computer, and then enter the project name (for example, myFirstProject). Click the **Save** button to create the project files.

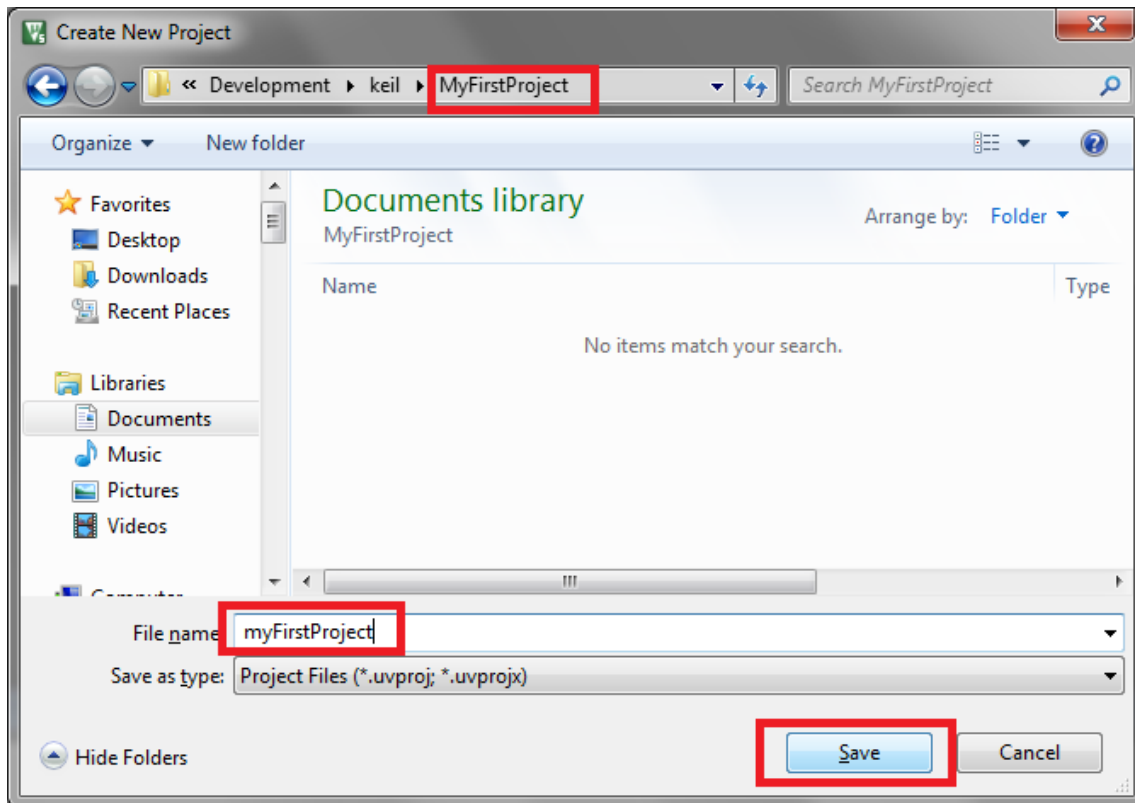


Figure 7. Assign Project Name

After the project has been created, μ Vision IDE prompts you to select the device to use, so that it can apply device-specific settings to the project. Go to **Texas Instruments** → **MSP432xx Series** → **MSP432P401x** and select the appropriate device from the list of available devices (see [Figure 8](#)).

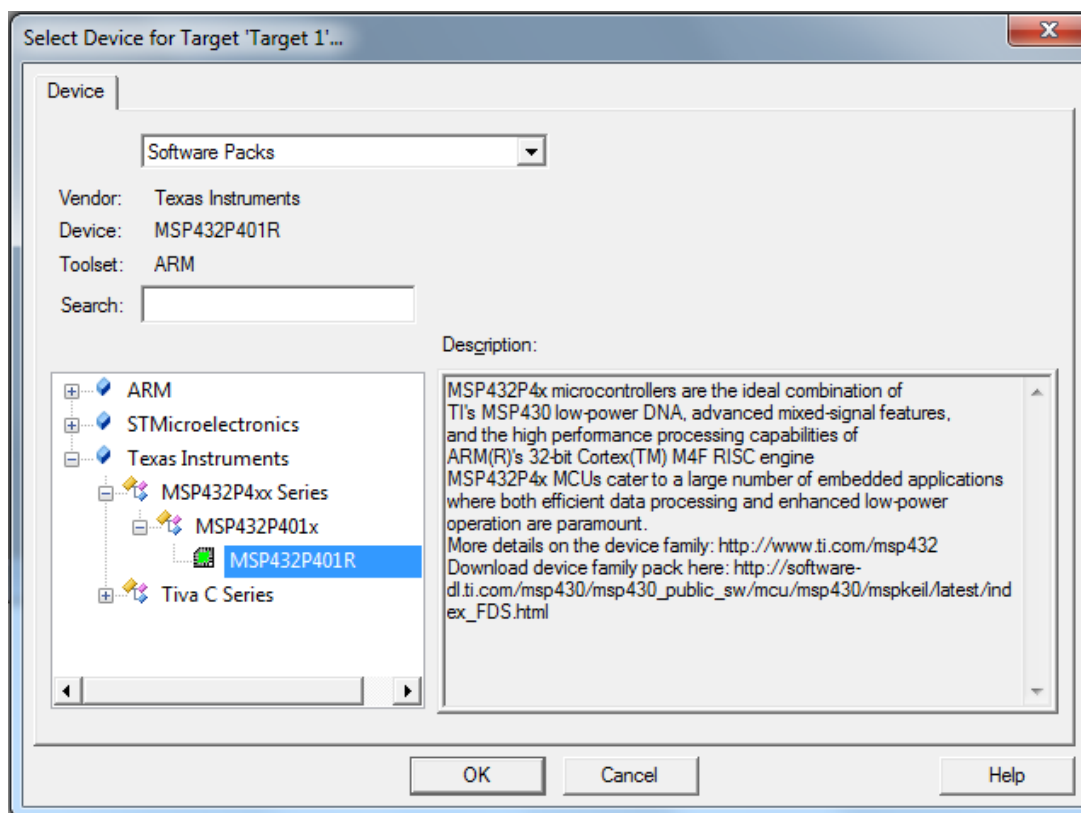


Figure 8. Select MSP432P401R Device

After the device has been selected, the run-time environment components can be added to the project (see [Figure 9](#)). If this step is skipped, the components can be added at a later point, or when additional components have been added through specific Software Packs, through the **Manage Run-Time Environment** button in the µVision IDE.

Run-Time Environment components include device startup and header files, and Cortex-M generic functions and definitions that are part of the Arm CMSIS package.

To develop code for the MSP432 MCUs, the selections for **CMSIS** → **CORE** and **Device** → **Startup** need to be enabled. Others are optional.

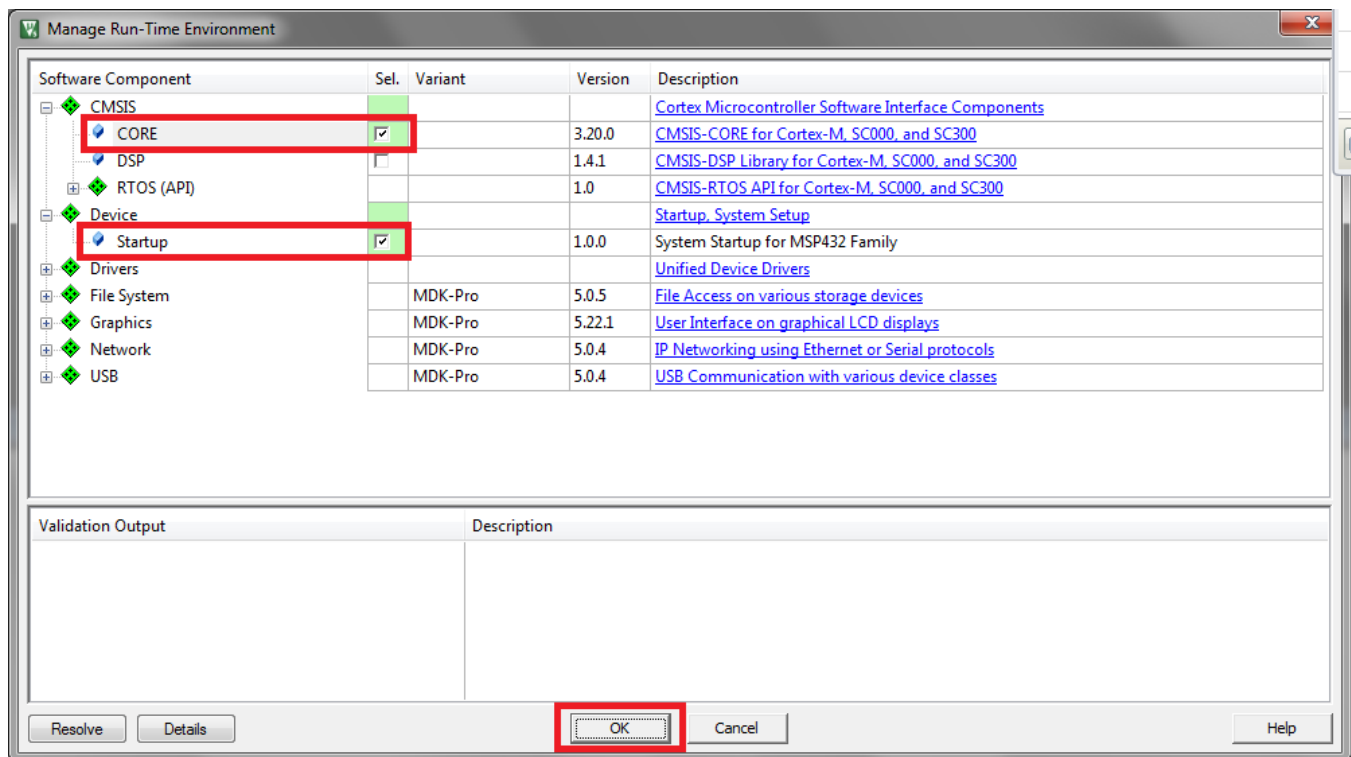


Figure 9. Add Run-Time Environment Components to the Project

When you have finished the wizard by clicking the **OK** button, μ Vision will show the newly created project containing two source code files:

- startup_MSP432p401r_uvision.s: Assembler file containing the interrupt vector table and a definition of all available interrupt service routines, including the reset vector
- system_MSP432p401r_uvision.c: C file containing low-level device initialization functions, including a watchdog timer halt command

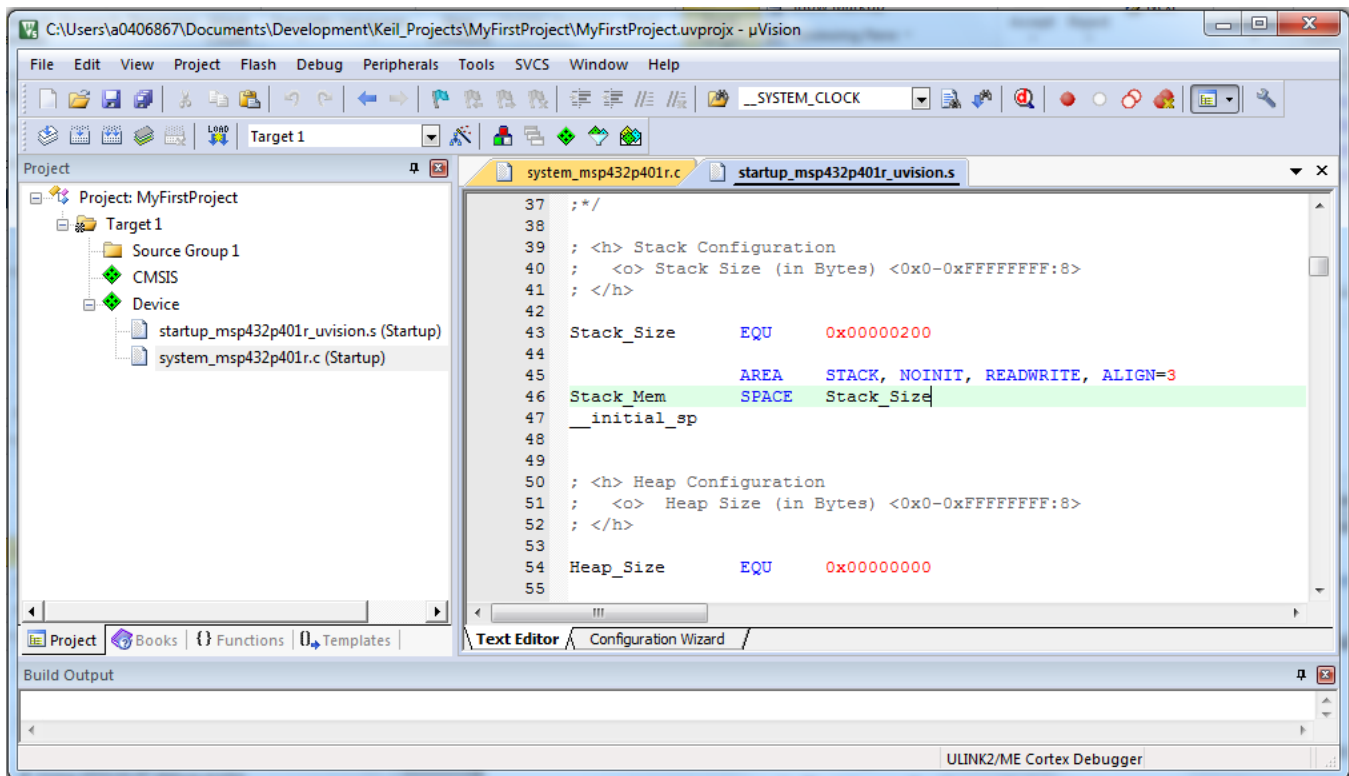


Figure 10. Empty Project

To start code development, a main() function is required. To create a source code file, right-click on **Source Group 1**, then select **Add New Item to Group** (see Figure 11).

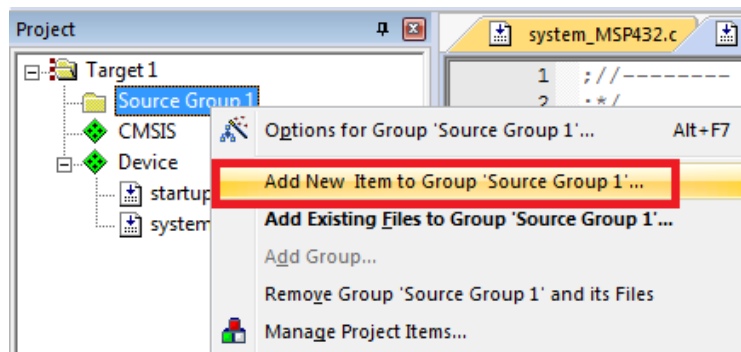


Figure 11. Add New Item to Project

A wizard opens in which several types of files can be created. Along with the MSP432 Device Family Pack, User Code Templates have been provided to select between the BlinkLED code and empty main code. The templates include the msp.h family header file and use C header files that provide MSP430 backward compatibility.

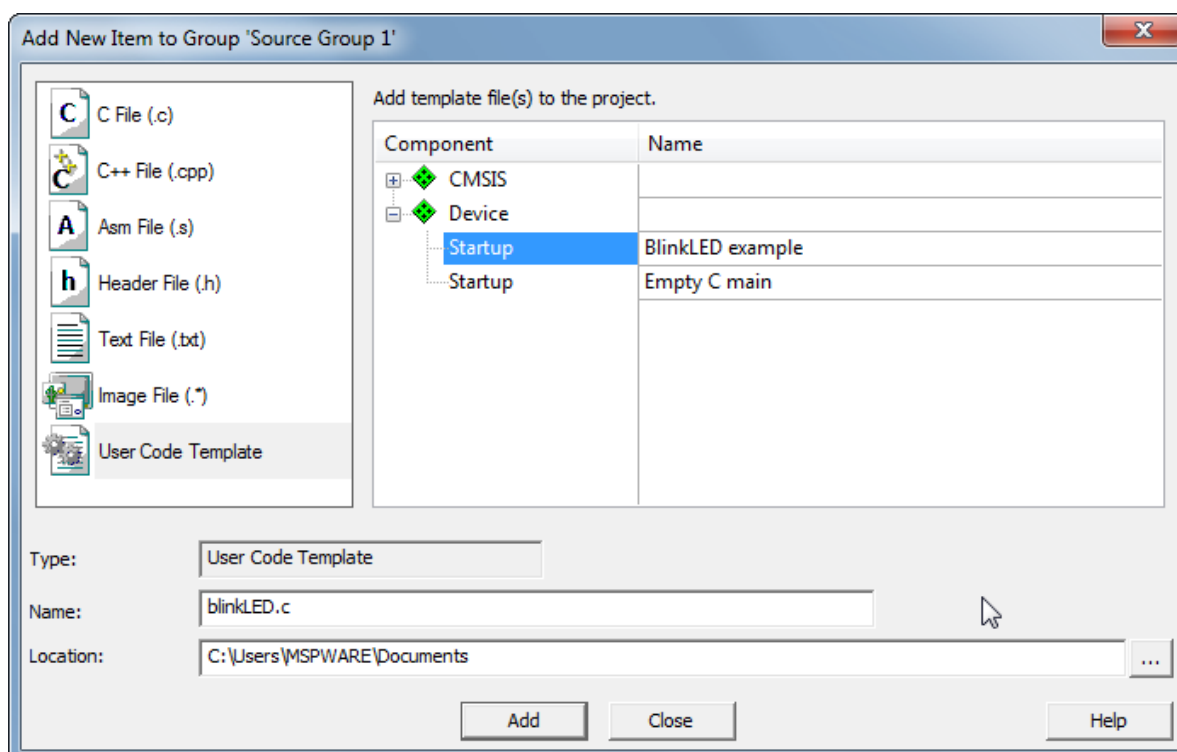


Figure 12. Add User Code Template

After adding a user code template to the code, the project is ready for compilation. Right-click on the project and select **Rebuild all target files** from the context menu, or press **F7** to rebuild all target files. If the project has been set up correctly, a compiler and linker output similar to the one shown in [Figure 13](#) is displayed.

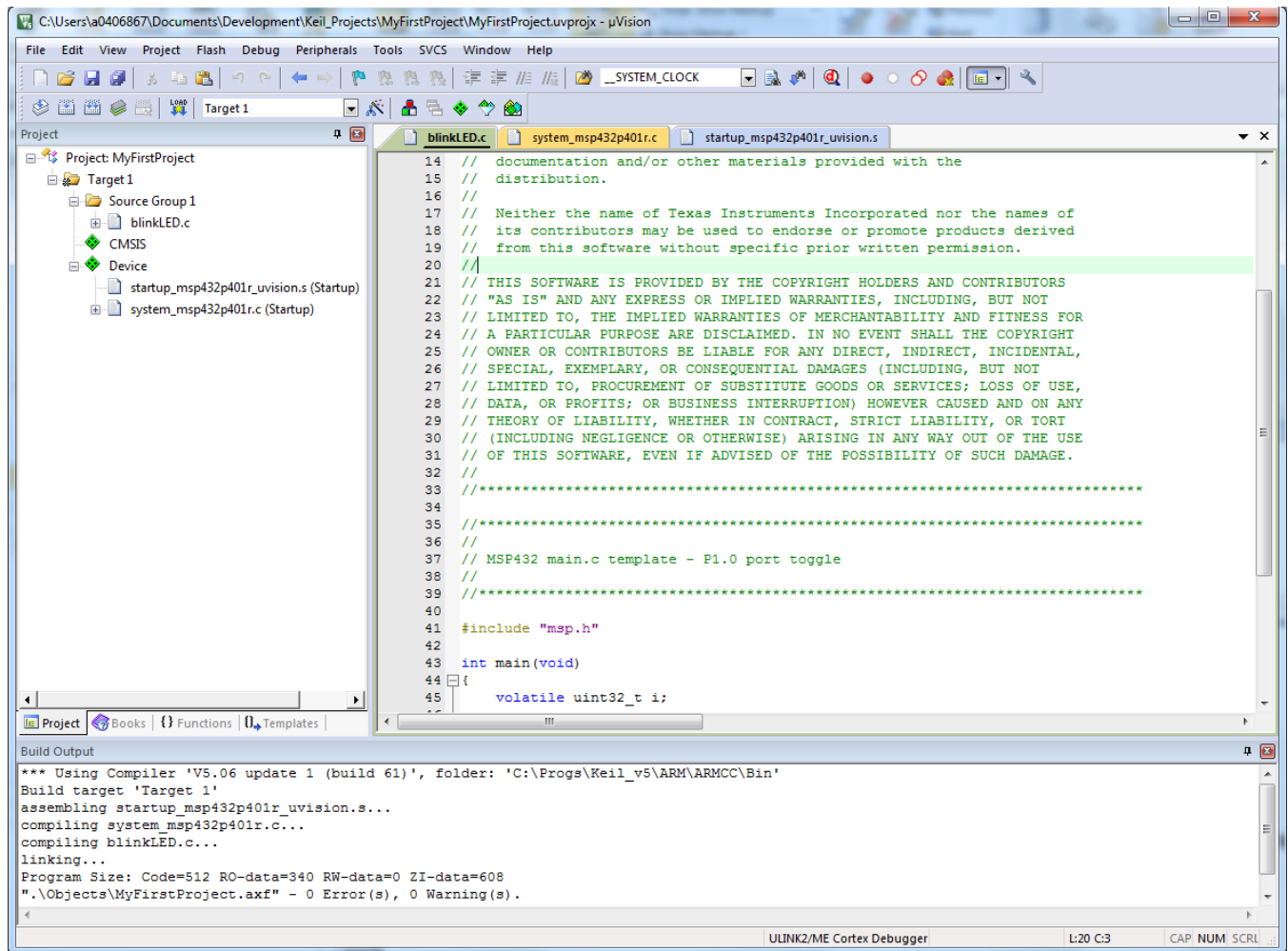


Figure 13. Compiling the Project

The project is now ready for upload to the device.

4 Opening an Existing SimpleLink MSP432 MCU Project

To open a project first launch the μ Vision IDE. After the IDE launches click **Project** → **Open Project** (see [Figure 14](#)).

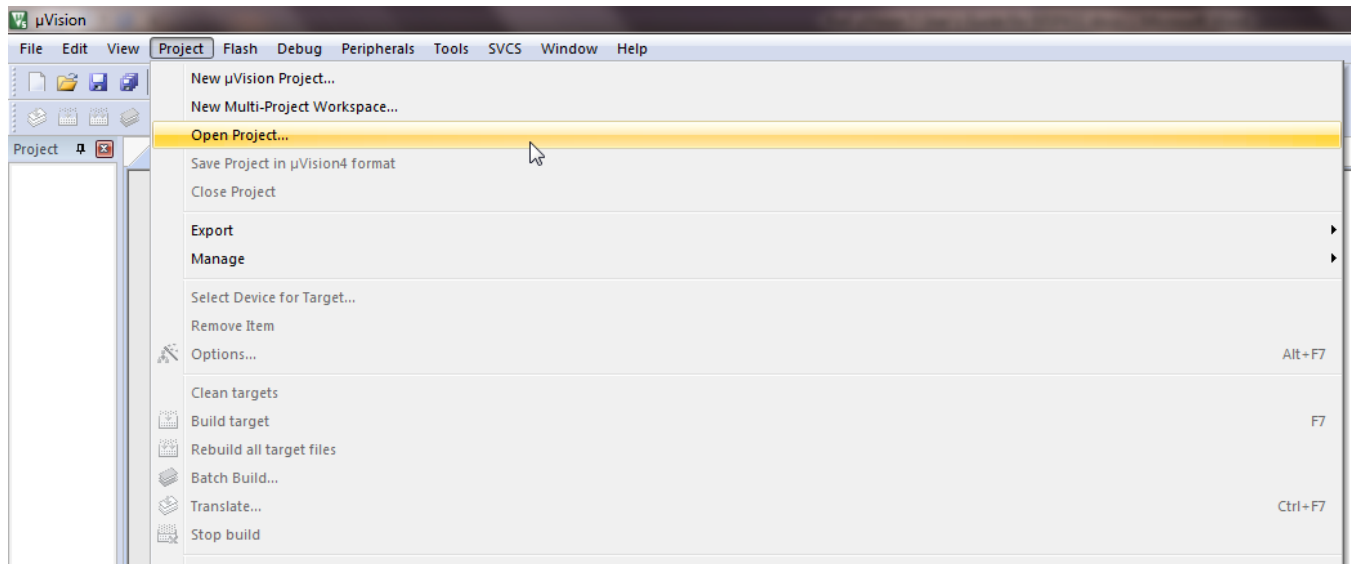


Figure 14. Opening a Project

After clicking "Open Project...", navigate to where the project is stored and double click the .uvprojx file to open the project in the IDE (see [Figure 15](#)).

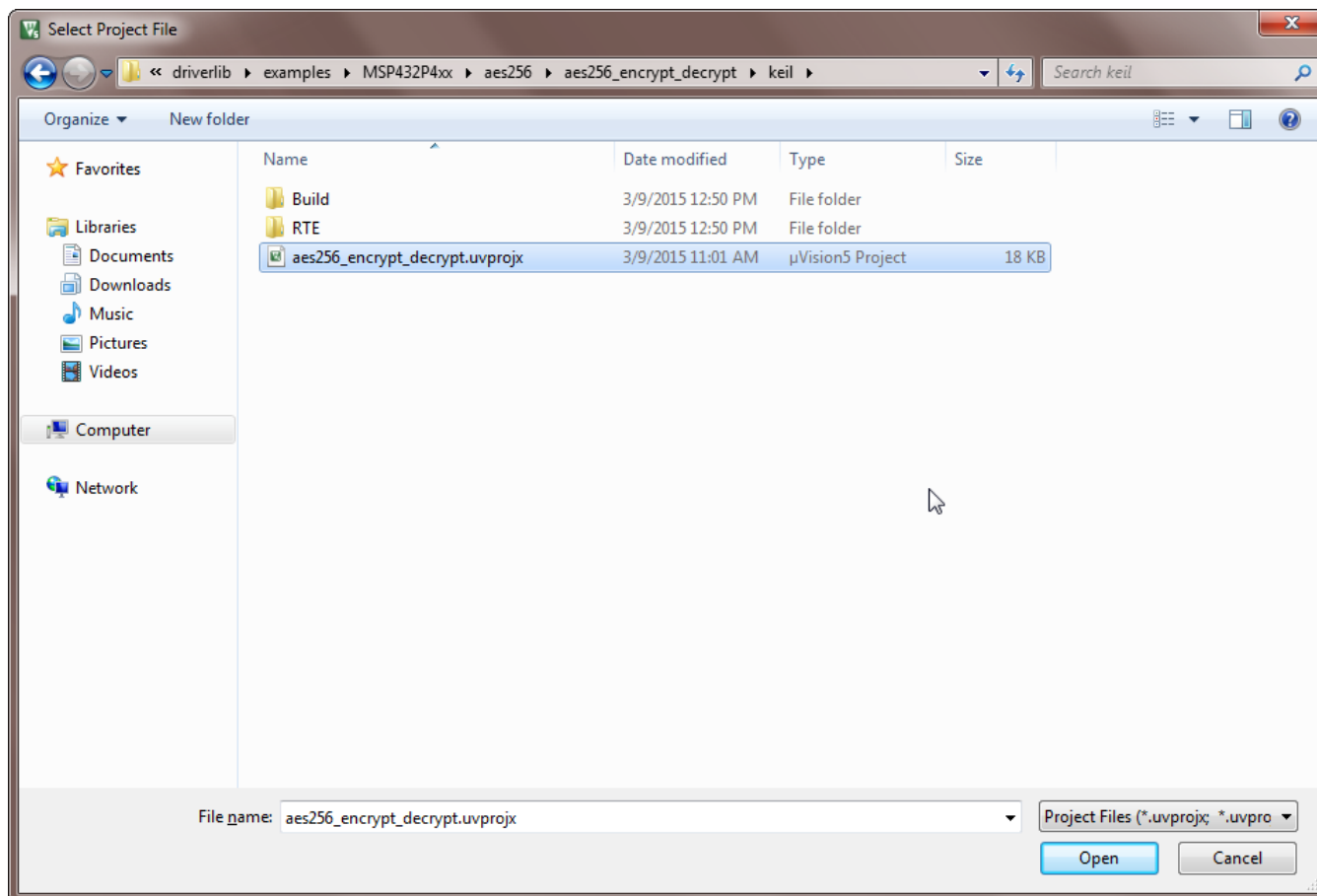


Figure 15. Choosing a Project to Import

5 Using SimpleLink MSP432 MCU Examples From the Software Development Kit

The SimpleLink MSP432 software development kit (SDK) contains many software examples, projects, documentation, application notes, and training for all MSP432 devices. This includes example projects for Keil µVision that work with MSP432 MCUs. For more information visit the [SimpleLink MSP432 SDK page](#).

All documentation for the SDK can be found in the SDK installation path in the docs folder. Open Documentation_Overview.html from that folder and then navigate to the Quick Start Guide for the IDE.

Each example in the SDK includes a folder named "Keil" that contains the *.uvprojx file with the project description. To import the project into the Keil MDK, copy it to the working location and open the project file (see [Figure 16](#)).

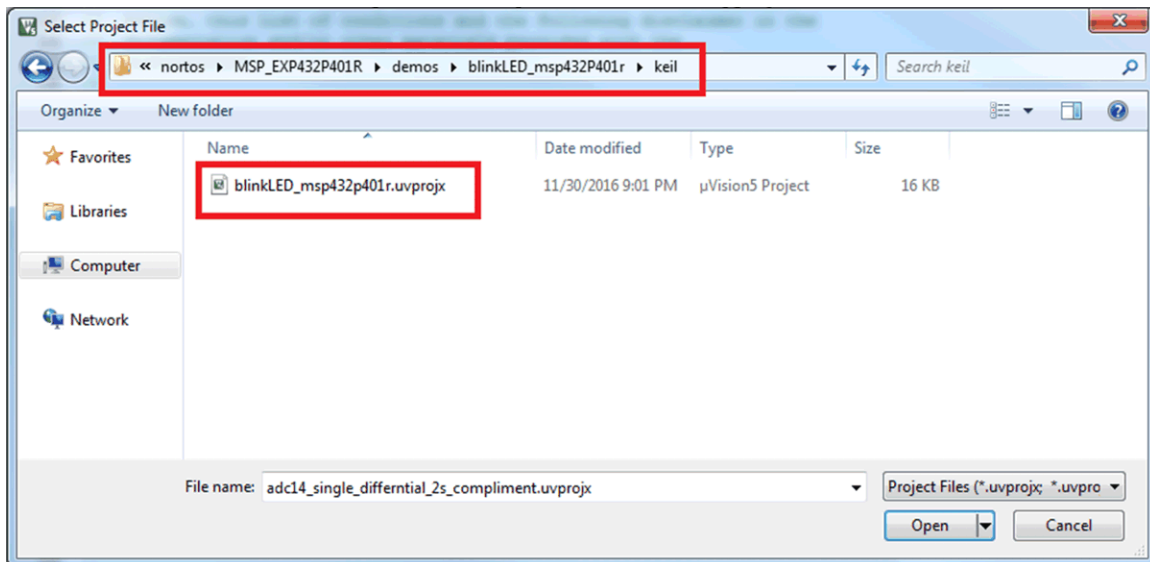


Figure 16. Select Project From SDK

You might need to change the device settings and debugger to fit the environment.

6 Debugging the Application

The following debug probes have been tested successfully with MSP432 MCUs and µVision IDE.

- Segger J-Link
- Keil ULINK2 and ULINK Pro
- XDS110-ET (using CMSIS-DAP or native support with MDK 5.20 and higher)

To use a debug probe that is not listed here, check with the vendor of the debug probe or with Arm if you experience problems.

6.1 Using Segger J-Link Debug Probe

To use the Segger J-Link debug probe, right-click on the active project, then select **Options for Target** (see Figure 17).

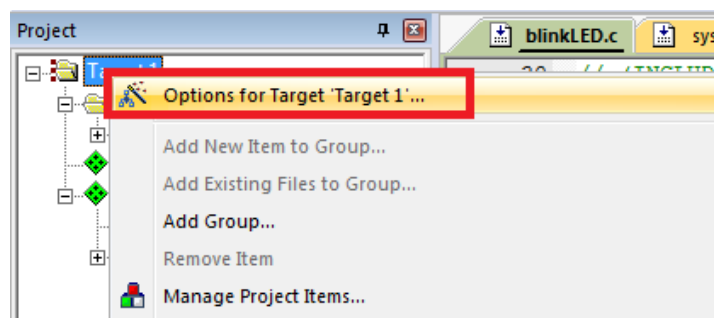


Figure 17. Project Options

In the **Target Options** window, select the **Debug** pane (see Figure 18). From the pulldown menu on the right-hand side, select the J-Link debug probe.

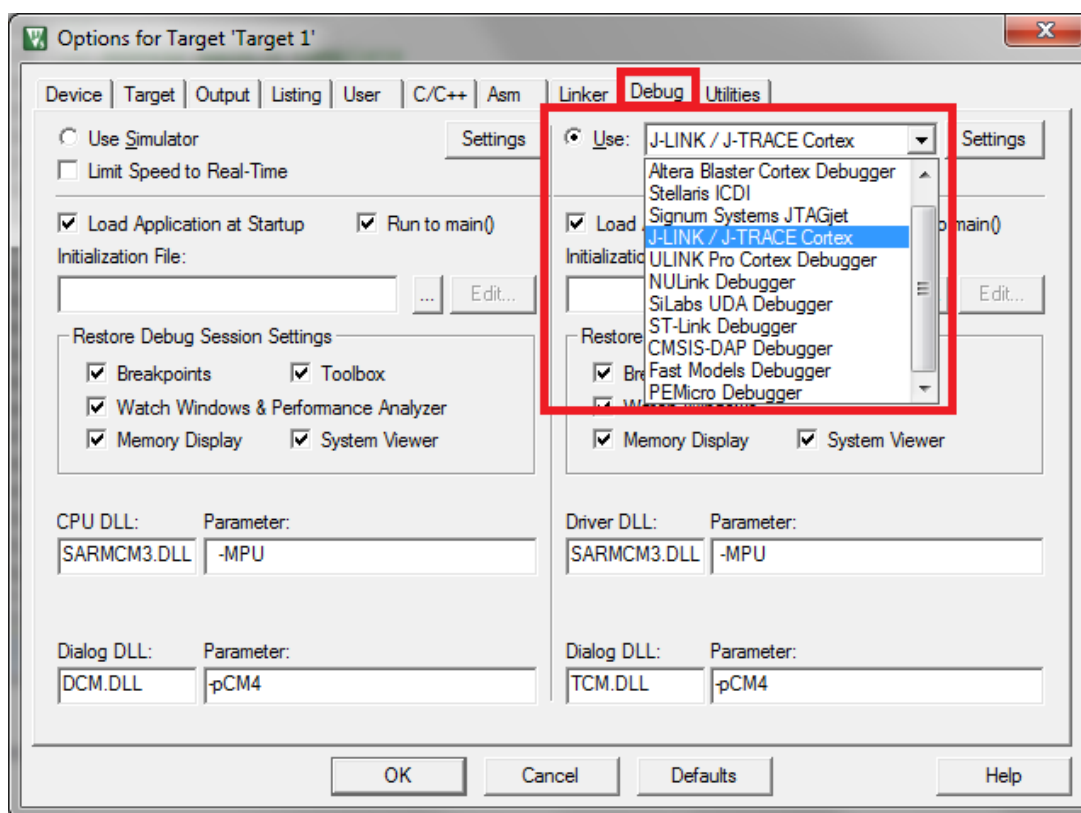


Figure 18. Select J-Link Debug Probe

Next, go to the debug probe specific settings by clicking on the **Settings** button next to the debug probe (see Figure 19).

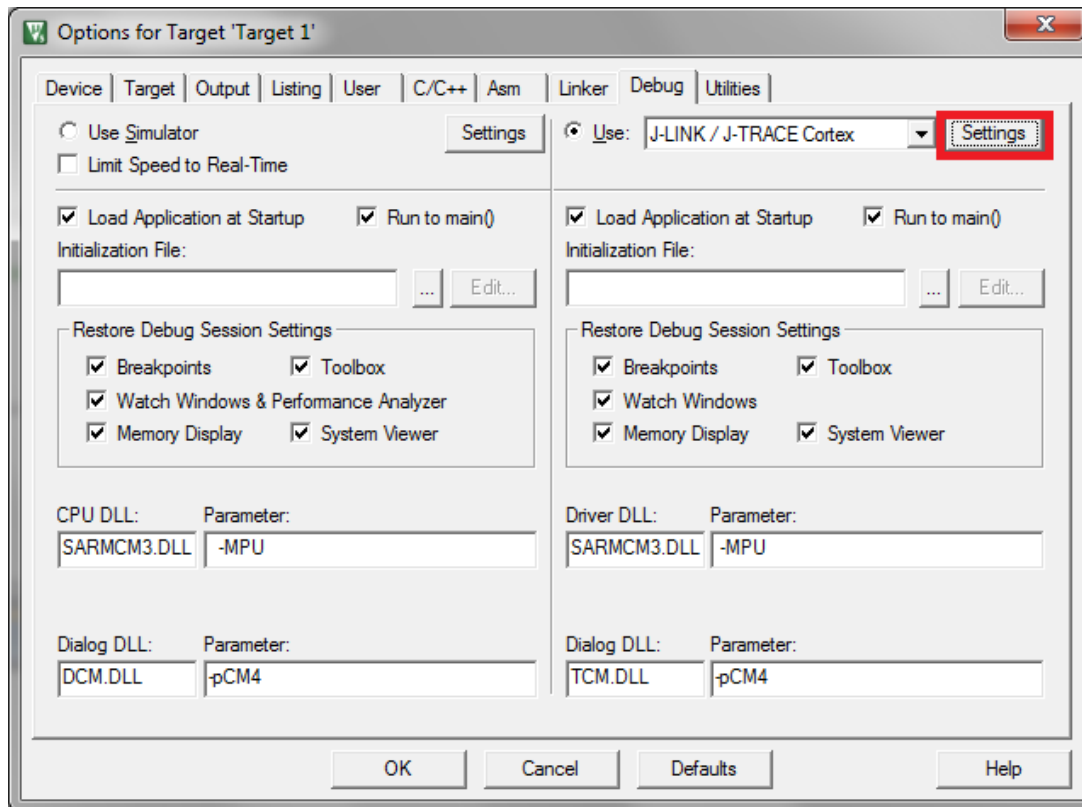


Figure 19. Go to J-Link Debug Probe Settings Window

In the Debug Probe Settings window (see [Figure 20](#)), make sure that both check boxes in the **Download Options** box are checked.

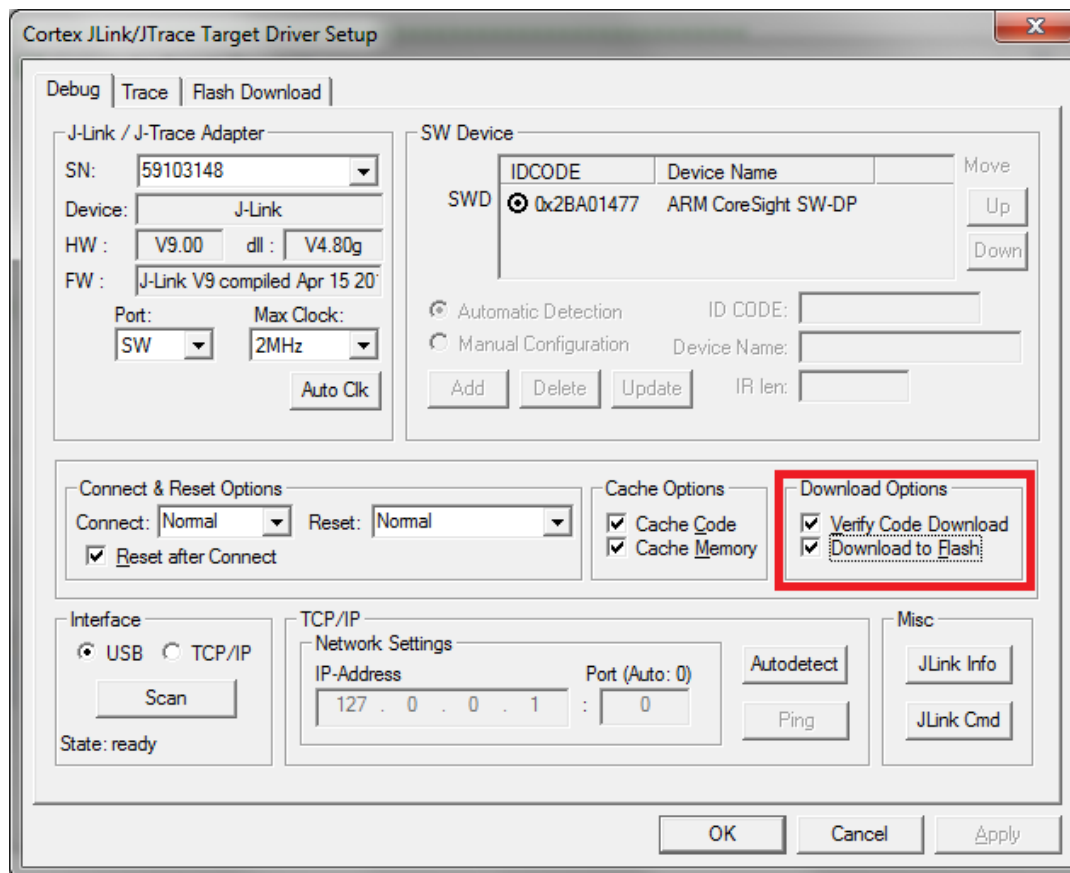


Figure 20. Enable J-Link Download Options

Go to the **Flash Download** pane to configure how the Flash memory should be erased, programmed, and/or verified. The settings shown in [Figure 21](#) issue a Flash mass erase prior to Flash programming and verify the programmed image before enabling debug. Even though the "Erase Full Chip" is selected, the bootstrap loader (BSL) is not erased. To replace the BSL, choose "Erase Sectors" and place code in the BSL section of memory.

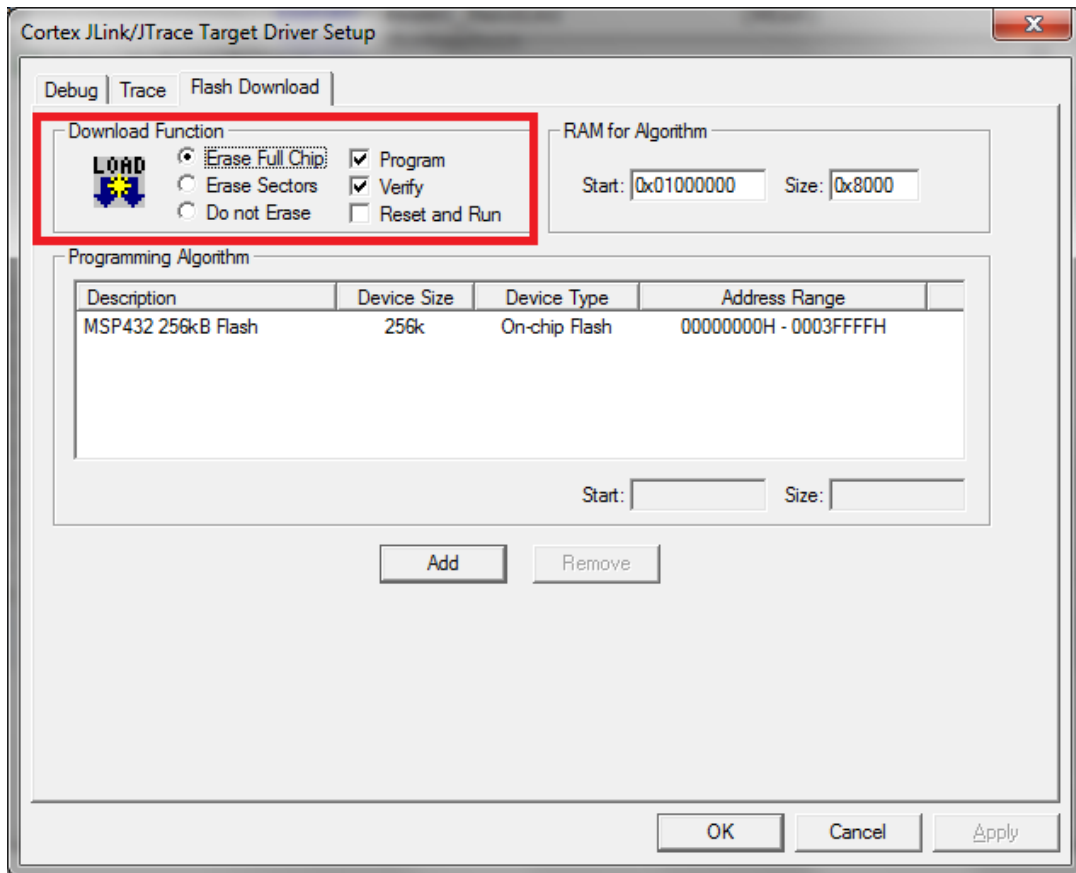


Figure 21. J-Link Download Function Configuration

After the Flash download has been configured, the Debug Session can be started. Go to **Debug** → **Start/Stop Debug Session**, or alternatively press **Ctrl+F5**.

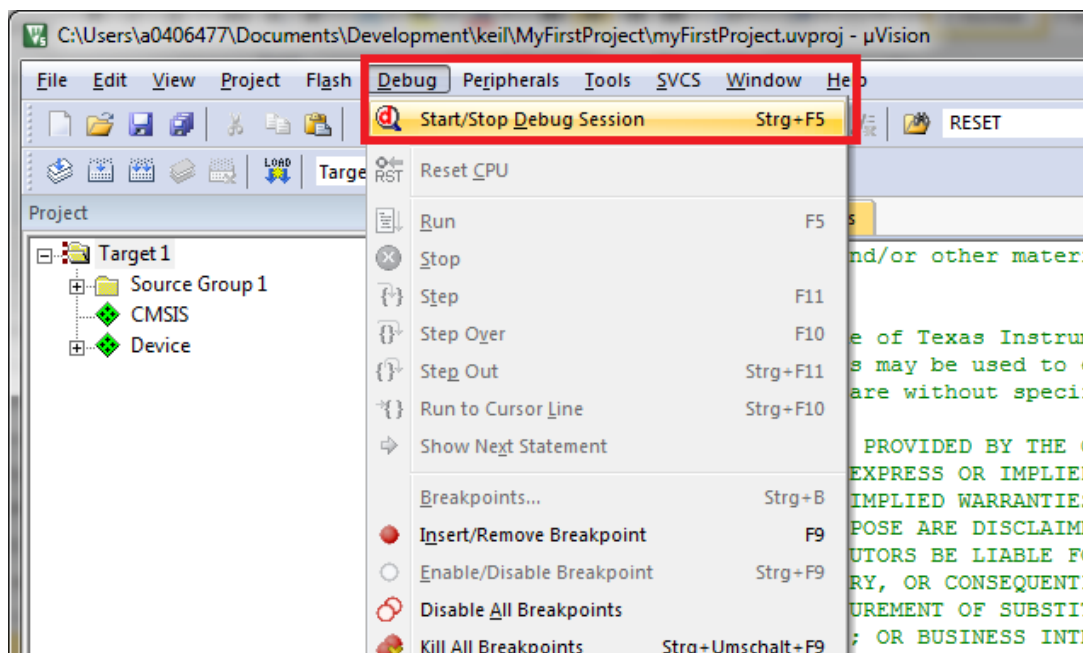


Figure 22. Start Debug Session

If using an older version of the Segger J-Link software, a popup might show once and indicate that the MSP432 device is not yet in the Segger J-Link database. It is safe to answer the manual device selection question with "No", but consider checking with Segger for an updated version of the J-Link software. After that, the debug session is started. [Figure 23](#) shows the Command window output during the start of the debug session.

```
Load "C:\\Users\\a0406477\\Documents\\Development\\keil\\MyFirstProject\\myFirstProject.axf"
* JLink Info: Device "UNSPECIFIED" selected (0 KB flash, 0 KB RAM).
Set JLink Project File to "C:\\Users\\a0406477\\Documents\\Development\\keil\\MyFirstProject\\JLinkSettings.ini"
* JLink Info: Device "UNSPECIFIED" selected (0 KB flash, 0 KB RAM).

JLink info:
-----
DLL: V4.80g, compiled Feb 13 2014 20:50:02
Firmware: J-Link V9 compiled Apr 15 2014 19:08:28
Hardware: V9.00
S/N : 59103148
Feature(s) : GDB

* JLink Info: Found SWD-DP with ID 0x2BA01477
* JLink Info: Found Cortex-M4 r0p1, Little endian.
* JLink Info: FPUUnit: 6 code (BP) slots and 2 literal slots
* JLink Info: TPIU fitted.
ROMTableAddr = 0xE00FF003

Target info:
-----
Device: MSP432P401R
VTarget = 3.262V
State of Pins:
TCK: 0, TDI: 0, TDO: 0, TMS: 1, TRES: 1, TRST: 0
Hardware-Breakpoints: 6
Software-Breakpoints: 8192
Watchpoints: 4
JTAG speed: 2000 kHz

Full Chip Erase Done.
Programming Done.
Verify OK.
```

Figure 23. Messages During Debug Session Startup

After the program image has been written to flash memory, the debug session starts and the debugger halts at the main function (see [Figure 24](#)).

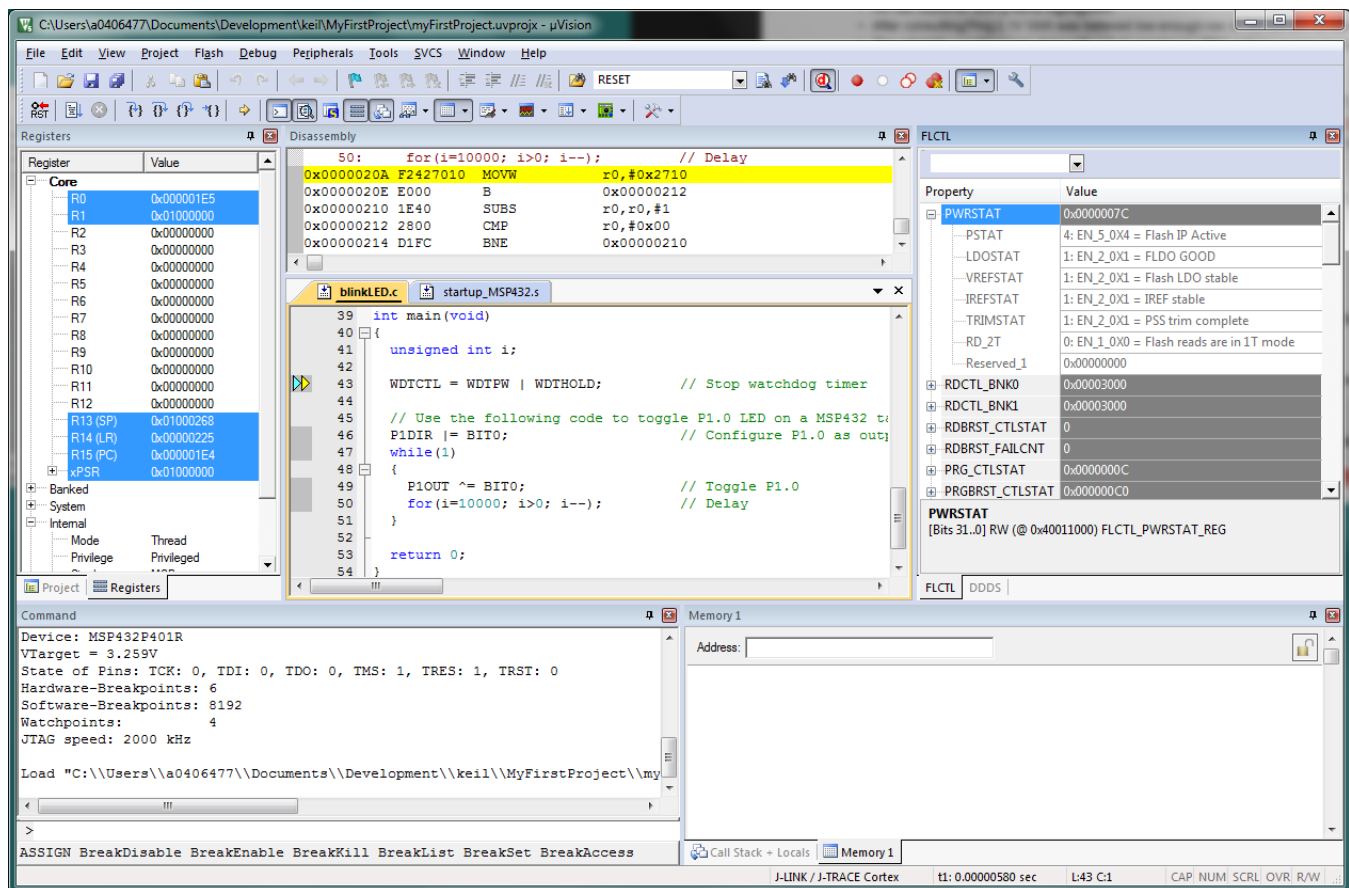


Figure 24. Debug Session Halted at main()

When using an MSP432 target socket board, you can benefit from the 5-V voltage output the J-Link provides on pin 19 of its Cortex-M debug connector to power the MSP432 device. This option needs to be enabled through the J-Link Commander, a console application available from Segger, since it is not available from the IDE. After it is enabled, the debug probe provides 5 V to the target system. See the [MSP432™ SimpleLink™ Microcontrollers Hardware Tools User's Guide](#) for instructions on how to configure the target socket board to use the 5-V power supply to generate 3.3-V device voltage, and see Segger's documentation for instructions on how to enable the voltage output.

Figure 25 shows the effect of the **power on** command, when applied in the J-Link Commander. Before executing the command, the measured target voltage is 0 V, and right after applying target power, 3.3 V is available as target voltage.

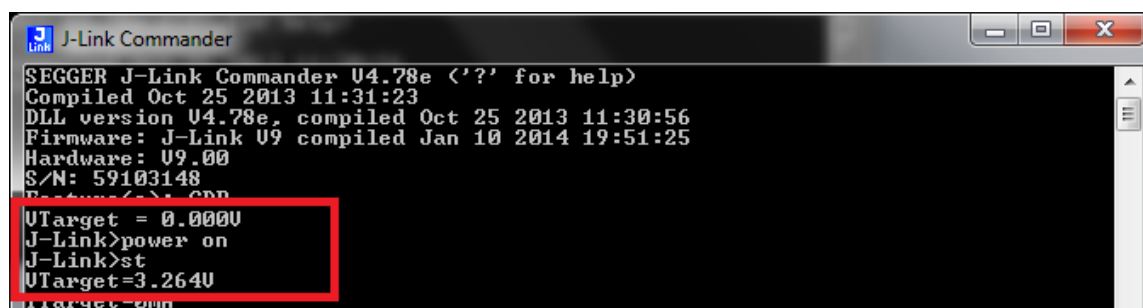


Figure 25. Using J-Link Commander to Enable Power Output to Target System

Now, download the program normally and debug using the Segger J-Link debug probe with uVision.

6.1.1 Working With Device Security (MSP432P4xx Devices Only)

If JTAG access is disabled on the device or the application needs to unlock a secure IP zone, a J-Link Script must be added to the debug session to enable a factory reset. During a debug session launch the J-Link control panel by clicking on the J-Link icon in the status bar (see [Figure 26](#)).

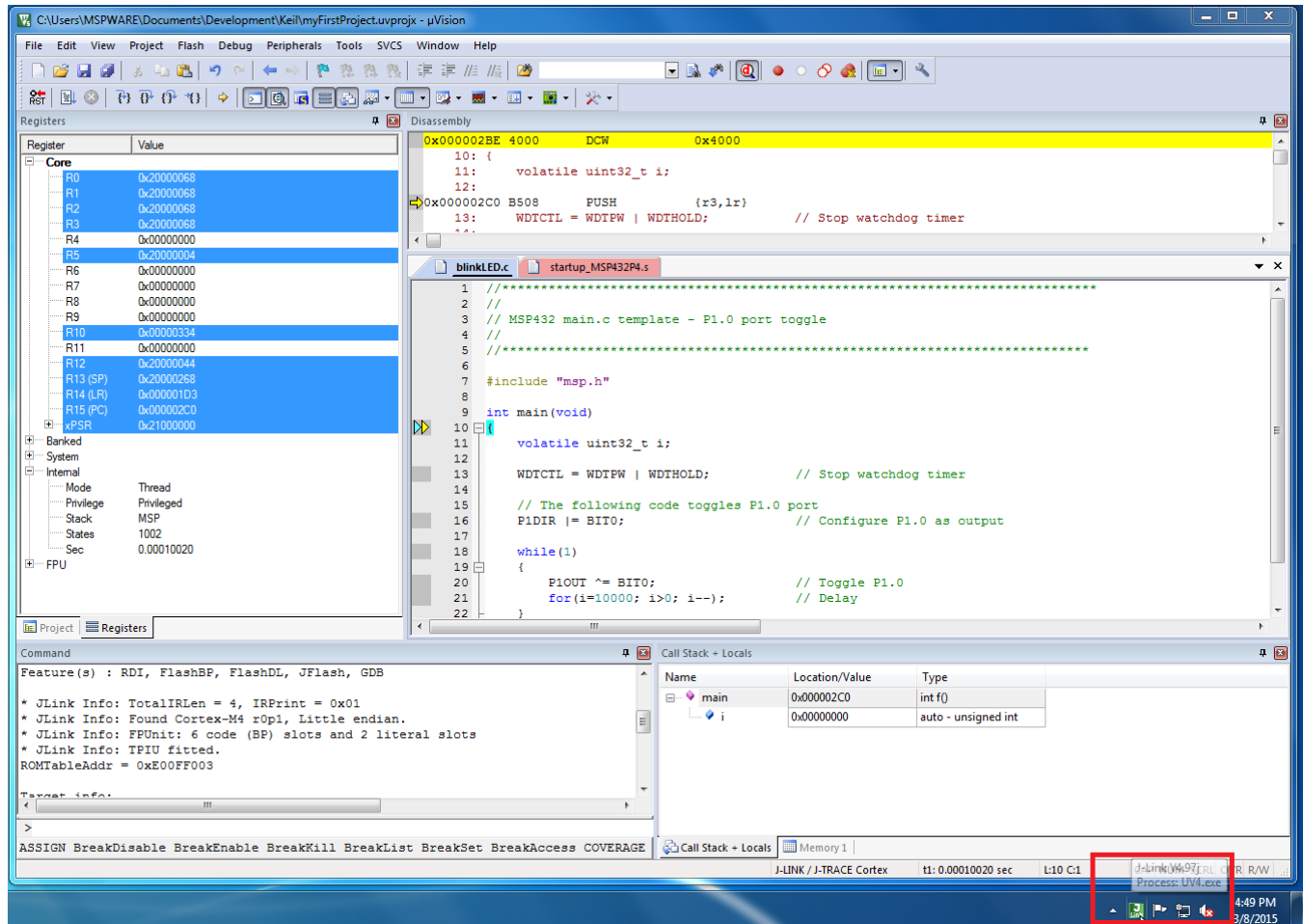


Figure 26. Launch J-Link Control Panel

After launching the J-Link control panel, add the J-Link script for MSP432 MCUs provided in the Keil pack. [Figure 27](#) shows the location of the script file.

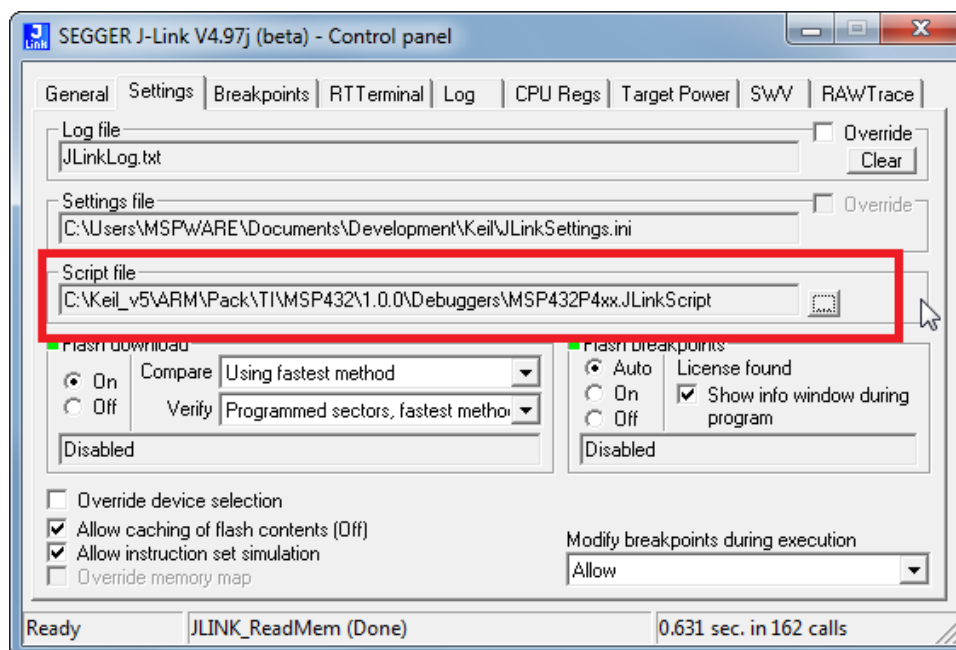


Figure 27. J-Link Control Panel

The J-Link script now runs at the launch of very debug session and every time code is downloaded to the device. If the device has been secured when trying to download code, a dialog box reports that the device is secured and will be erased (see Figure 28).

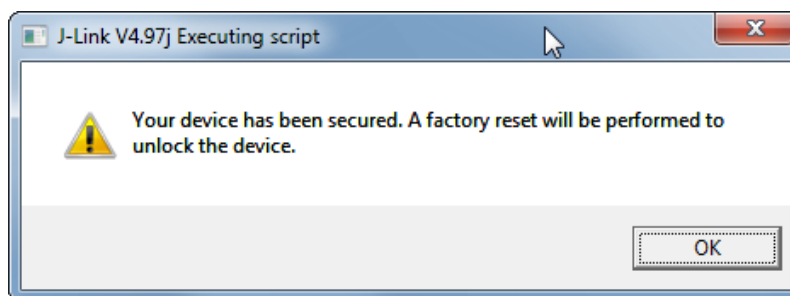


Figure 28. J-Link Script Detecting That the Device Has Been Secured

Click OK on the dialog box to issue a factory reset, which erases any code present on the device, and to start to download the compiled code. After the factory reset is a complete, a confirmation dialog box is displayed (see Figure 29).

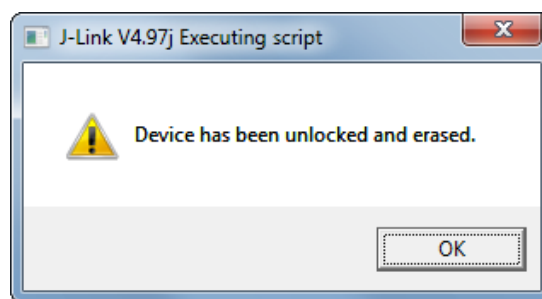


Figure 29. Device Has Been Unlocked and Erased

6.2 Using Keil ULINK2 and ULINK Pro debug probe

To use Keil ULINK2 or ULINK Pro debug probes, right-click on the active project, then select **Options for Target**.

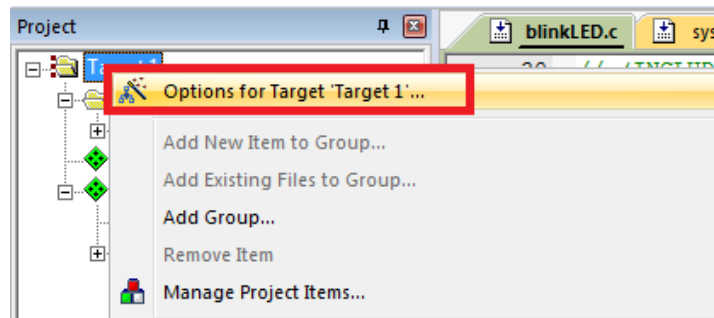


Figure 30. Project Options

In the Target Options window, select the **Debug** pane (see Figure 31). From the pulldown menu, select the ULINK2 or ULINK Pro debug probe. Be careful to select the right debug probe, as the driver is different.

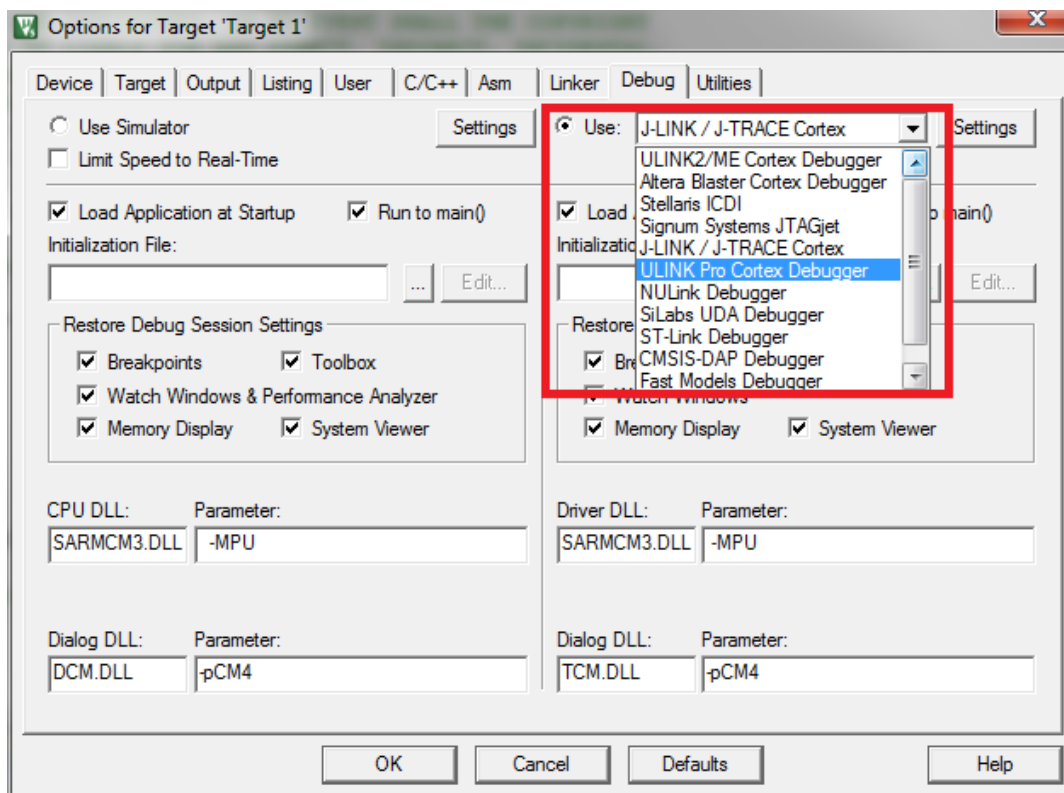


Figure 31. Selecting the ULINK Pro Debug Probe

In the Debug Probe Settings window (see Figure 32), make sure that both check boxes in the **Download Options** box are checked.

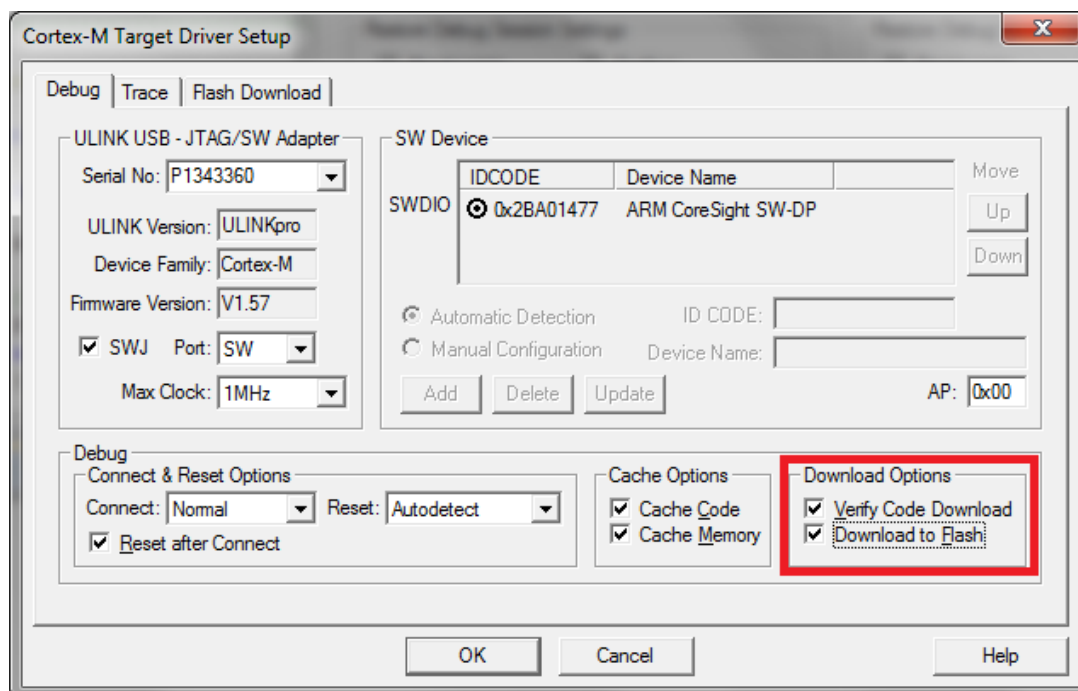


Figure 32. Selecting ULINK Pro Download Options

Now move to **Flash Download** pane to configure the way the Flash memory should be erased, programmed, and/or verified. [Figure 33](#) shows the settings to issue a flash mass erase prior to flash programming and verify the programmed image before enabling debug. Even though the "Erase Full Chip" is selected, the BSL is not erased. To replace the BSL, choose "Erase Sectors" and place code in the BSL section of memory.

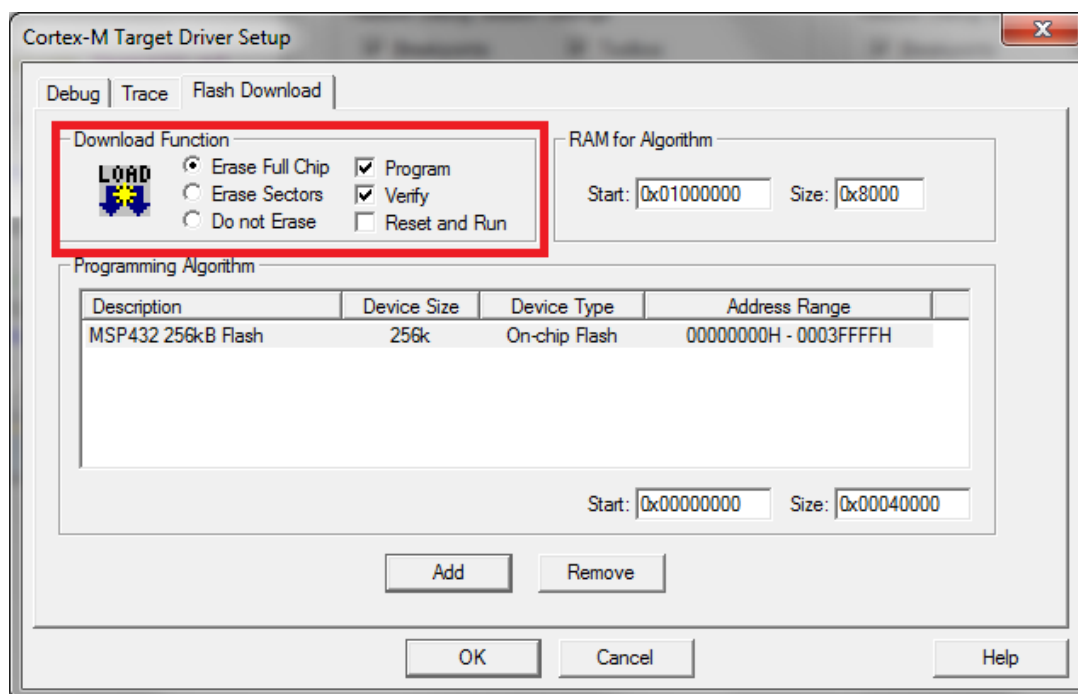


Figure 33. Select ULINK Pro Download Function

After the flash download has been configured, the debug session can be started. Go to **Debug** → **Start/Stop Debug Session**, or alternatively press **Ctrl+F5**.

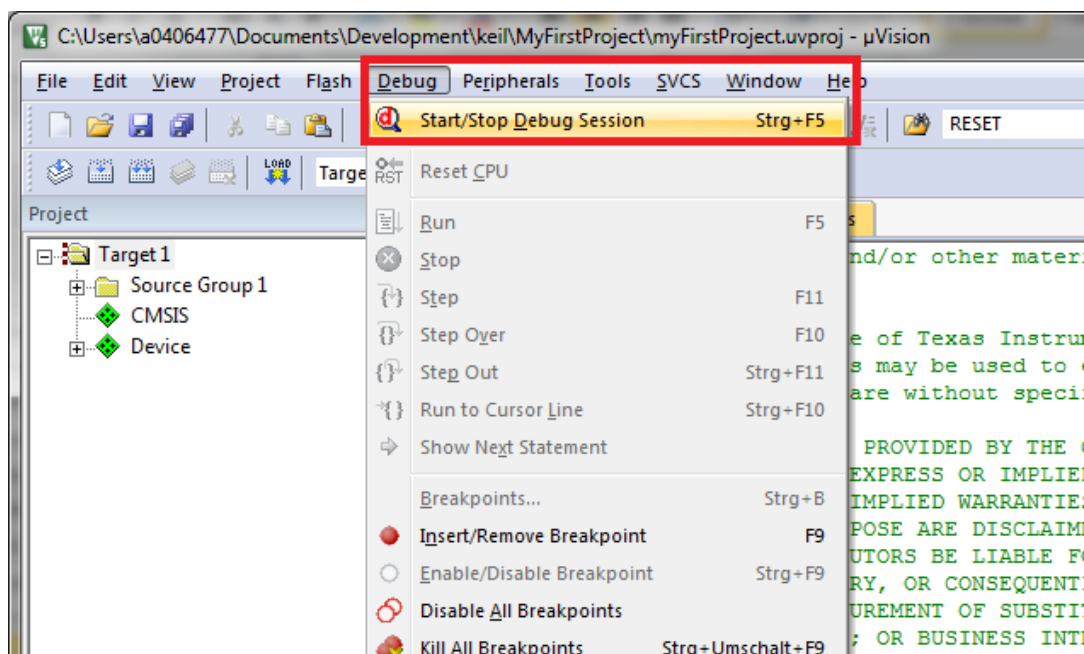


Figure 34. Start Debug Session

After the program image has been written to flash memory, the debug session starts and the debugger halts at the main function (see [Figure 35](#)).

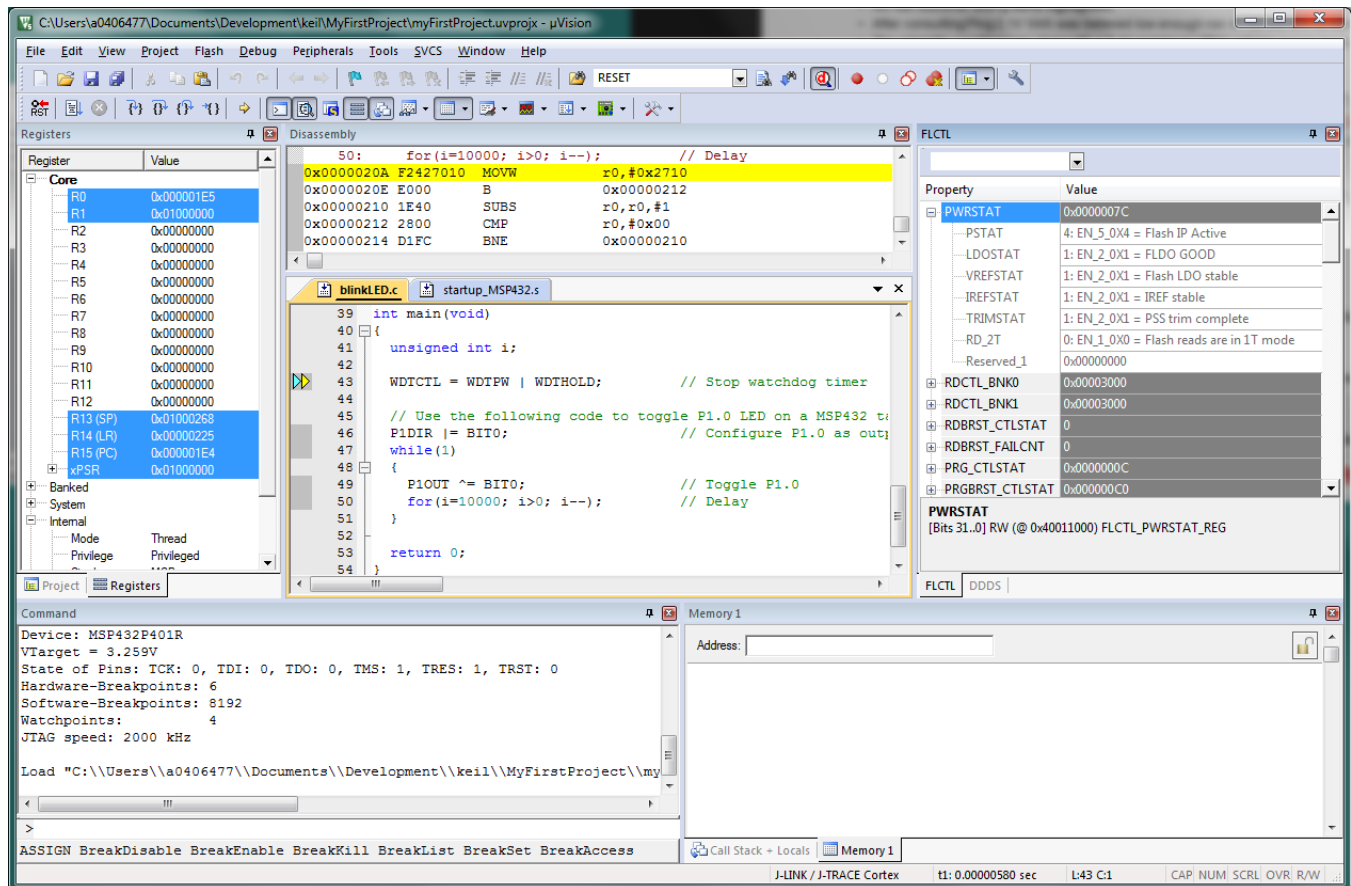


Figure 35. Debug Session Successfully Started and Halted at main()

6.2.1 Working With Device Security (MSP432P4xx Devices Only)

If JTAG access is disabled on the device or the application needs to unlock a secure IP zone, Keil automatically runs a check on the device before downloading code. If Keil finds that the device has been secured, a dialog box is displayed (see Figure 36).

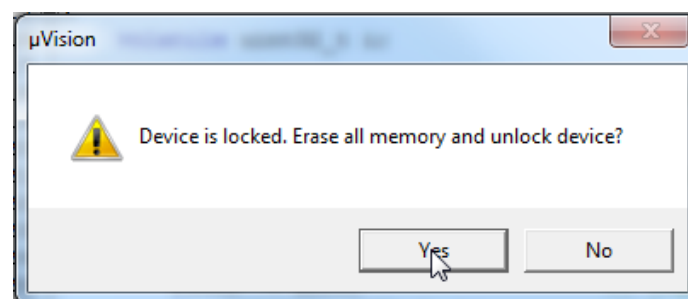


Figure 36. Dialog Box Asking to Perform a Factory Reset

Click **Yes** to perform a factory reset, which unlocks the device so that code can be downloaded. After the code has been downloaded, the debug session starts.

Click **No** to end the debug session and leave the device locked.

6.3 Using XDS110-ET Debug Probe

The XDS110-ET is the debugger featured on the MSP-EXP432P401R LaunchPad™ development kit and allows developers to quickly get started developing and debugging applications on MSP432 MCUs. In addition, the XDS110 is available as a stand-alone probe (see [XDS110 JTAG Debug Probe](#)).

To use XDS110-ET debug probe, right-click on the active project, then select **Options for Target**.

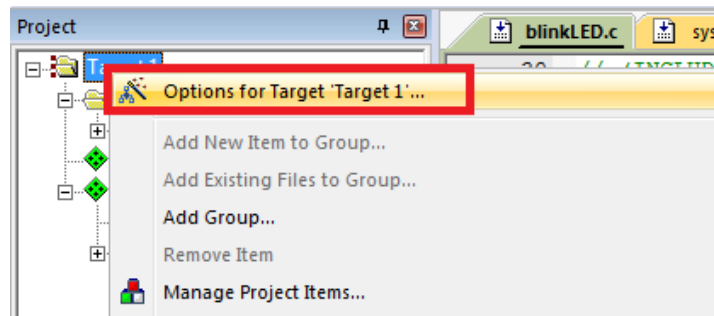


Figure 37. Project Options

In the Target Options window, select the **Debug** pane. From the pulldown menu, select either CMSIS-DAP Debugger or TI XDS Debugger. TI XDS gives higher performance and is available in MDK 5.20 or higher.

NOTE: TI XDS110 debuggers are supported natively or through CMSIS DAP. TI XDS native support in MDK 5.20 needs a separate installation of the TI emulation pack software in the default folder C:\TI\ccs-base.

MDK 5.20 needs version 6.0.83.1 of the emulation pack and the associated firmware 2.2.5.1 on the XDS110 debug probe. This can be checked and changed with the xdsdfu tool provided in the emulation pack. See the readme in the installation folder (C:\TI\ccs_base\common\uscif\xds110) for details.

All versions of the emulation pack are available in the [Texas Instruments XDS Emulation Software](#).

Higher versions of MDK update the XDS firmware accordingly.

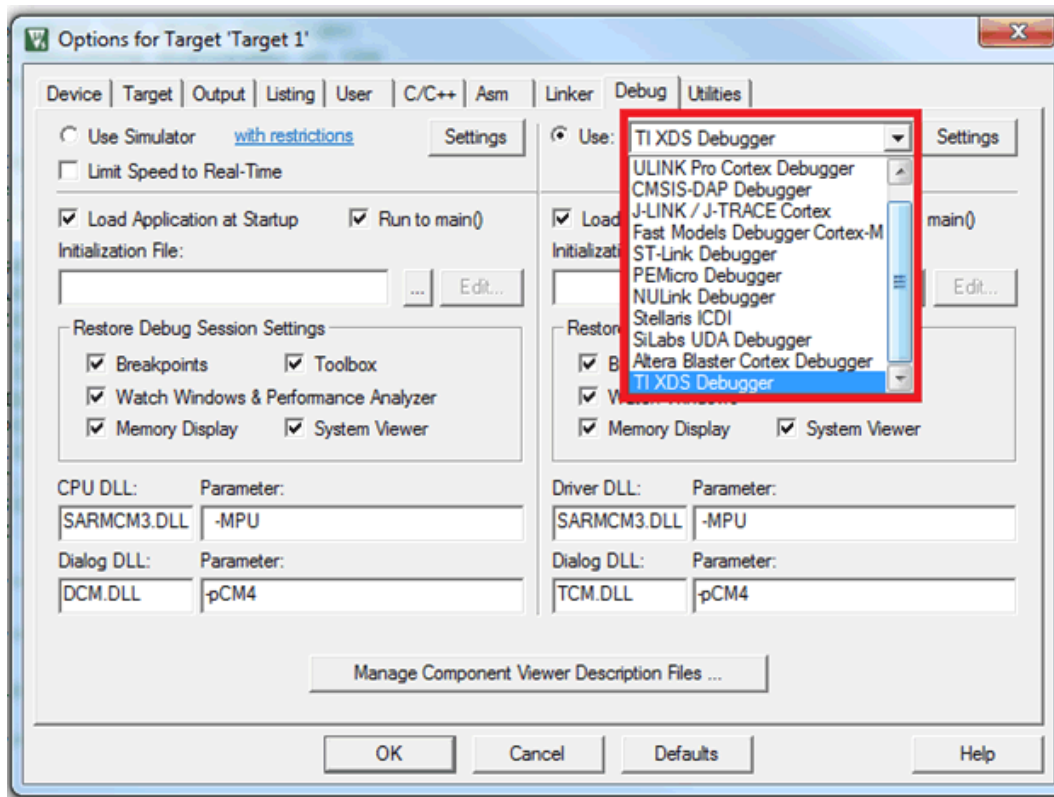


Figure 38. Selecting the XDS110 Debug Probe

In the Debug Probe Settings window (see Figure 39), make sure that both check boxes in the **Download Options** box are checked.

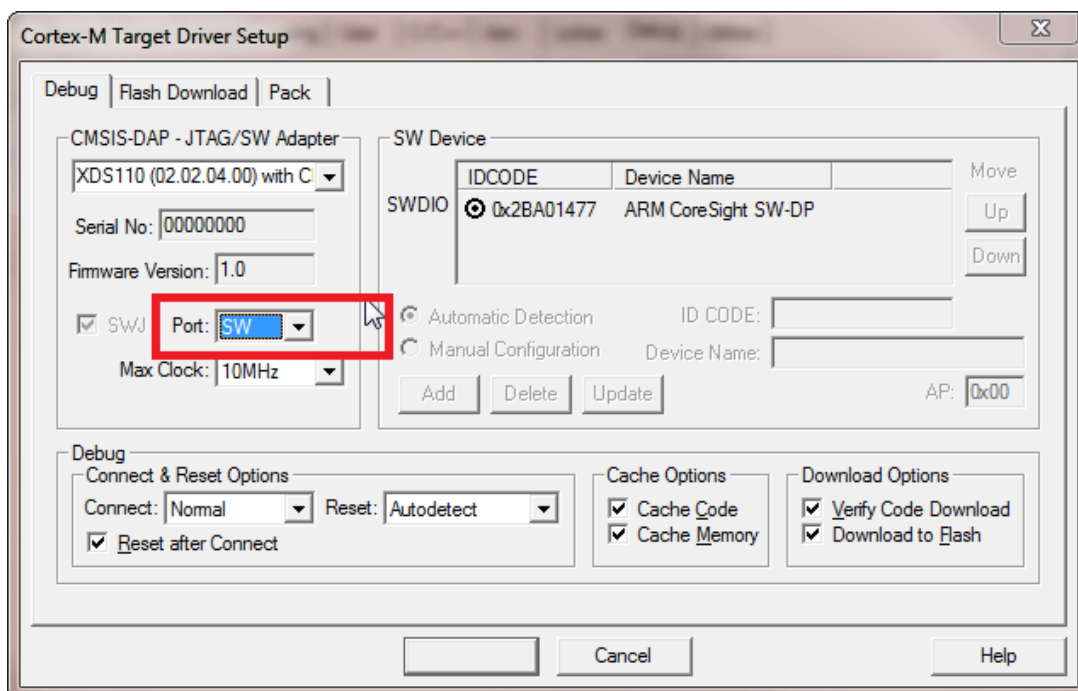


Figure 39. Selecting CMSIS-DAP Settings

Now move to **Flash Download** pane to configure the way the Flash memory should be erased, programmed and/or verified. The settings shown in Figure 40 issue a Flash mass erase prior to Flash programming and verify the programmed image before enabling debug. Even though the "Erase Full Chip" is selected the Bootstrap Loader (BSL) will not be erased. To replace the BSL choose "Erase Sectors" and place code in the BSL section of memory.

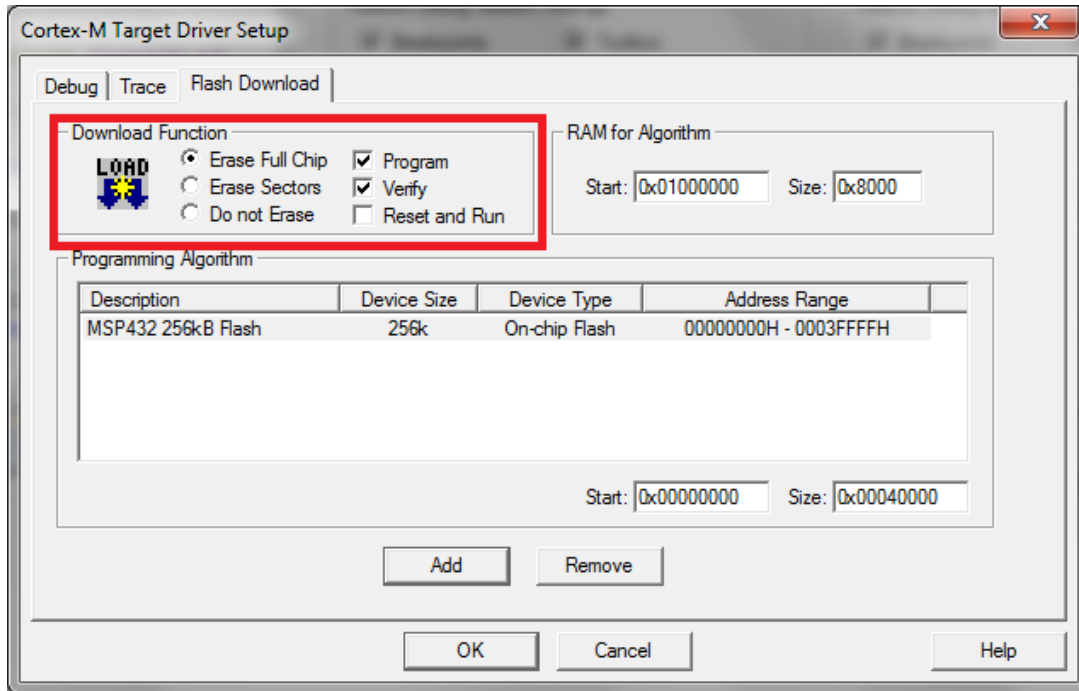


Figure 40. Select ULINK Pro Download Function

After the Flash download has been configured, the Debug Session can be started. Go to **Debug** → **Start/Stop Debug Session**, or alternatively press **Ctrl+F5**.

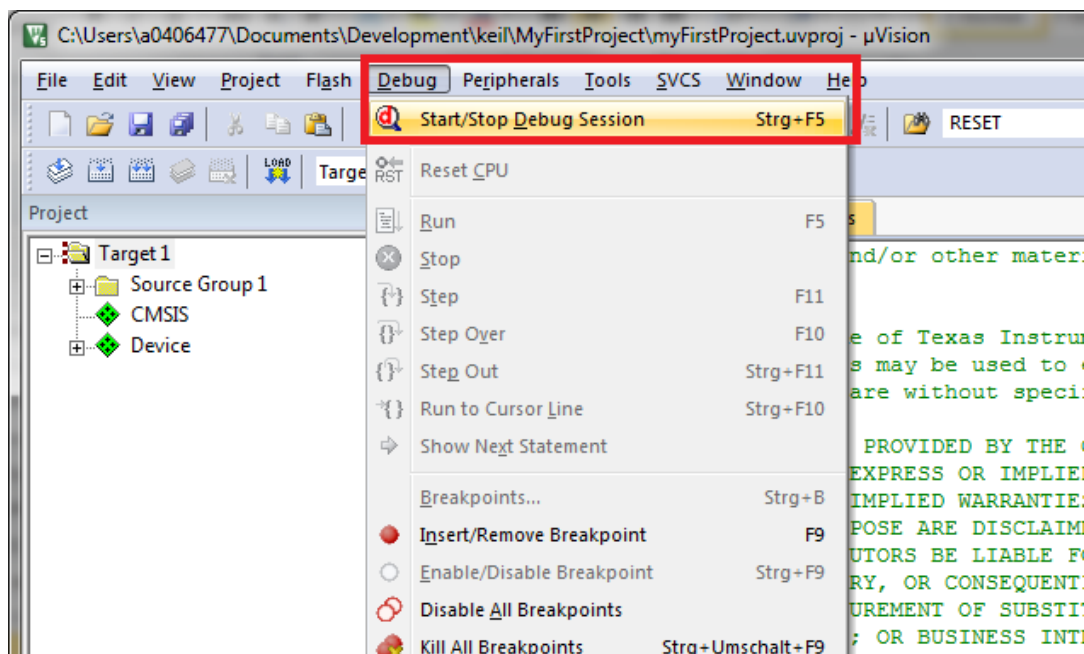


Figure 41. Start Debug Session

After the program image has been written to flash memory, the debug session starts and the debugger halts at the main function (see Figure 42).

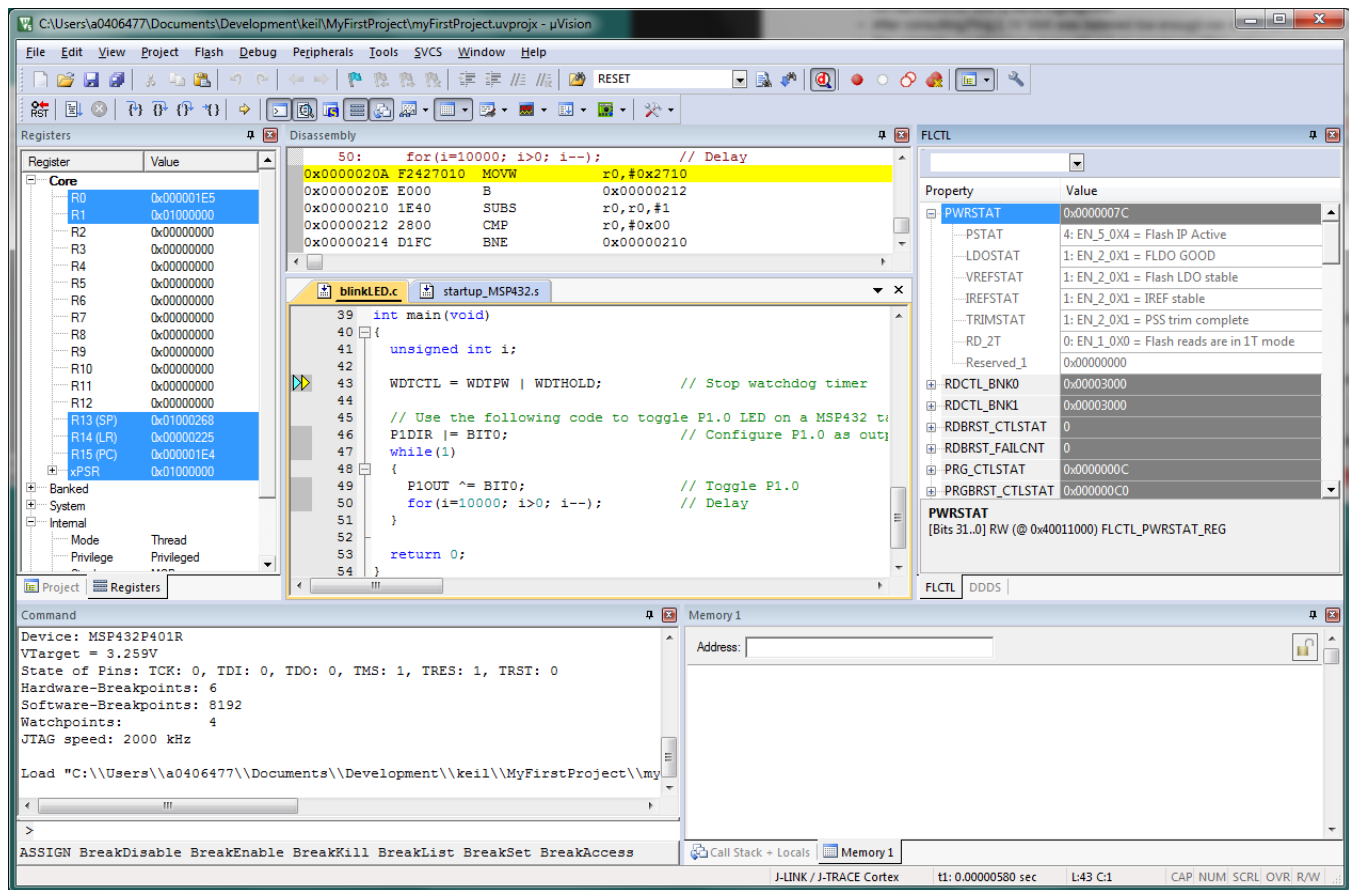


Figure 42. Debug Session Successfully Started and Halted at main()

6.3.1 Working With Device Security (MSP432P4xx Devices Only)

If JTAG access is disabled on the device or the application needs to unlock a secure IP zone, Keil automatically runs a check on the device before downloading code. If Keil finds that the device has been secured, a dialog box is displayed (see Figure 43).

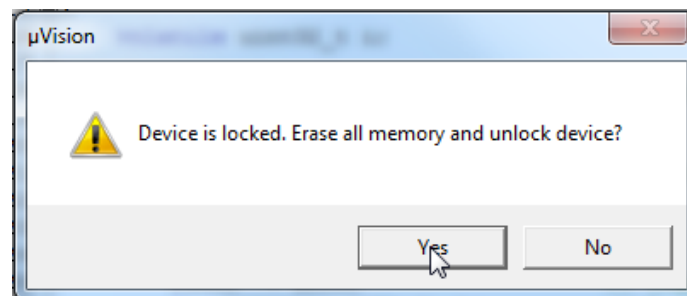


Figure 43. Dialog Box Asking to Perform a Factory Reset

Click "Yes" to perform a factory reset. The device is unlocked so that code can be downloaded. Click "No" to quit the debug session. The device remains locked. After the code has been downloaded, the debug session starts.

6.4 Debugging Driver Lib in ROM

The MSP432P4xx family includes a complete peripheral driver library (DriverLib) fully integrated into the ROM memory. Developers can leverage the ROM DriverLib for multiple benefits including access to highly robust and tested APIs, single-cycle ROM execution speed at lower power consumption, and freeing up memory space for additional application code. Developers can gain access to ROM APIs by adding DriverLib header file to projects and linking to a prebuilt library.

The driver library source code is now part of the SimpleLink MSP432 SDK. The driver library is a low-level software layer below the TI drivers, which are also part of the SDK. In the SDK, the DriverLib source code can be found in <SDK_InstallationPath>\source\ti\devices\msp432p4xx\driverlib.

For more information on MSP432P4xx Driver Library and what is provided in ROM DriverLib, see the documentation in the [SimpleLink MSP432 SDK](#).

6.4.1 Add ROM Source and Symbol to an Existing SimpleLink MSP432 MCU Project

All DriverLib example projects from the SDK already have ROM debugging enabled. To add ROM debugging to an existing project, perform the following steps:

1. Include the driver library header in the source code:
#include <ti/devices/msp432p4xx/driverlib/driverlib.h>
2. If the path to the driver library headers and sources is not yet included in the project, add it now. Right-click on the active µVision project, then select Options for Target. Add the include path to the driver library source folder in the C/C++ options (see [Figure 44](#)). For example, add "C:\TI\simplelink_msp432_sdk_1_20_00_45\source\".
3. Make sure that the correct preprocessor symbol for the device is defined (for example, __MSP432P401R__ for MSP432P401R devices).

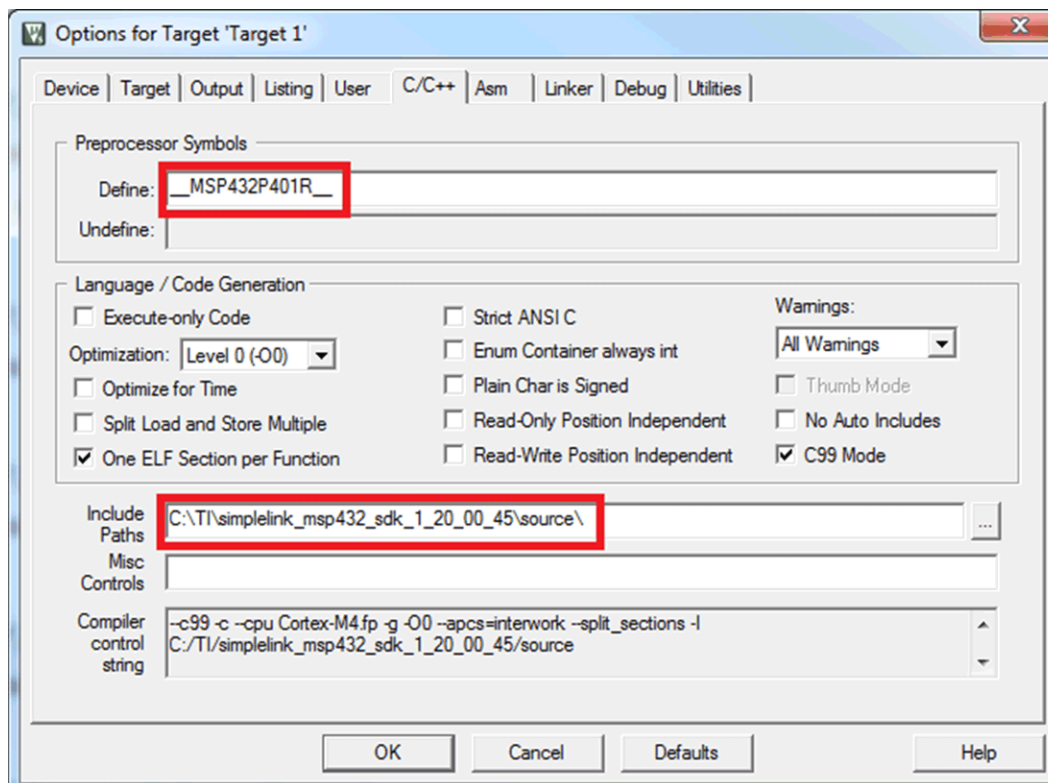


Figure 44. Adding Path to the Driver Library Folder

4. To add the precompiled driver library to the project, right-click on the Library folder (create one if it does not exist), then select Add Existing Files to Group 'Library' and add the file msp432p4xx_driverlib.lib from the subfolder "source\ti\devices\msp432p4xx\rom\keil" in the SDK directory. Make sure to select the file type Library file (*.lib) in the dialog.

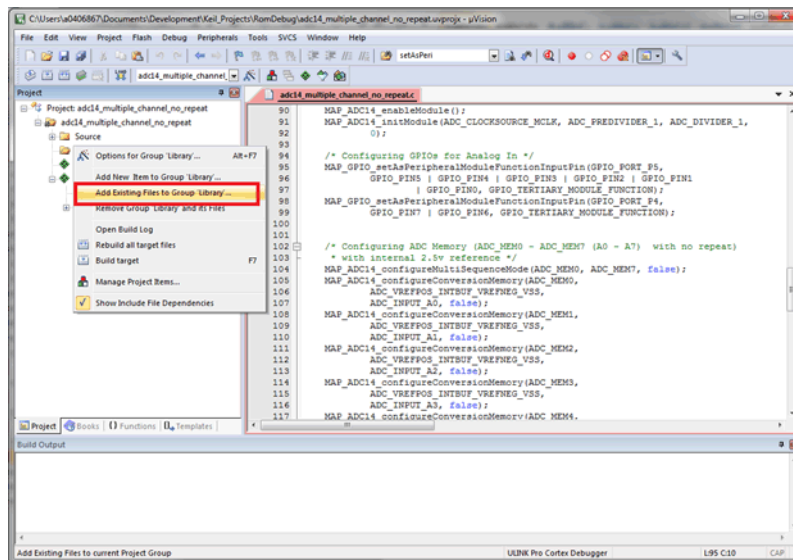


Figure 45. Adding Driver Library to a Project

6.4.2 Load ROM Symbol at the Beginning of a Debug Session

The loading of the ROM debugging symbols is done through an initialization file. Copy and adapt the "keil_rom_load.ini" found in the "\source\ti\devices\msp432p4xx\rom\keil" directory in the SDK folder, or create a new file for the project.

In either case, the file should contain a line with loading instructions and the absolute or relative path to the driver library rom image. This example assumes that the SimpleLink MSP432 SDK has been installed in "C:\TI" and, therefore, the initialization file would look like the following:

LOAD

"C:\TI\simplelink_msp432_sdk_1_20_00_45\source\ti\devices\msp432p4xx\rom\msp432_driverlib_rom_image.out" INCREMENTAL NOCODE

Right-click on the active μ Vision project and select Options for Target. Add the initialization file in the Debug section as shown in Figure 46.

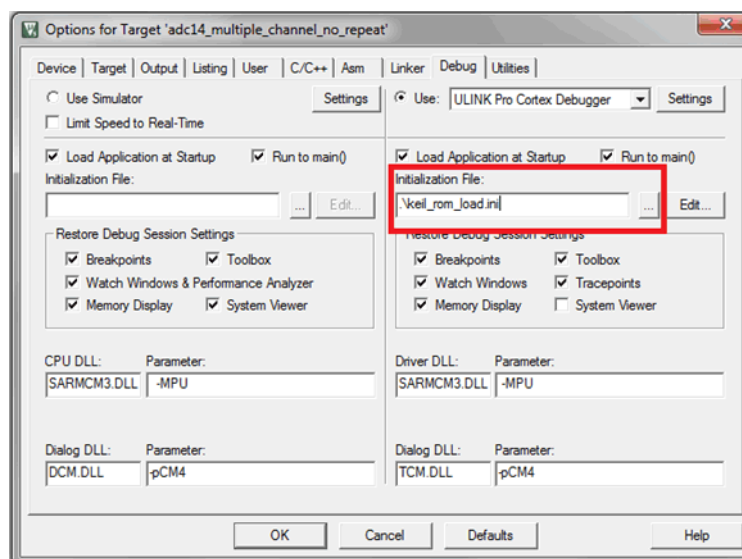


Figure 46. Adding Initialization File for Debugging

When the next debug session is started, you can step through the ROM API. The IDE automatically opens the corresponding source file (see [Figure 47](#)).

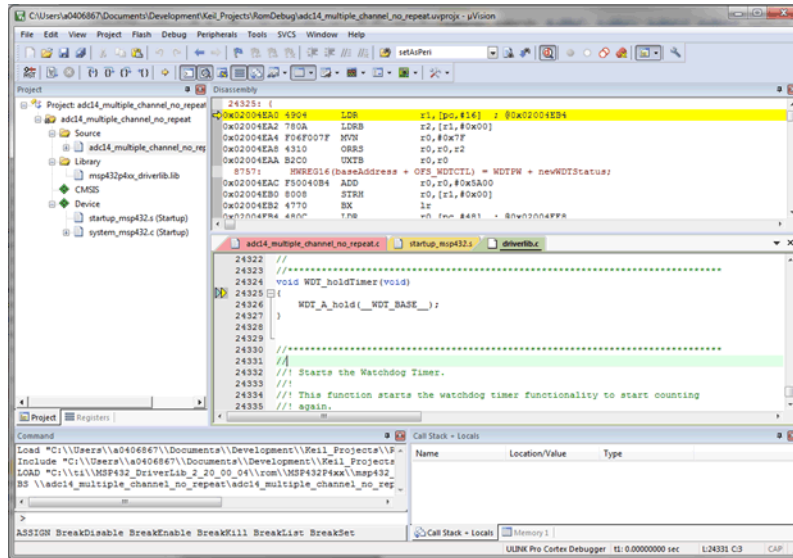


Figure 47. Stepping Through ROM Driver Lib Source Code

6.5 Using Serial Wire Output (SWO) Hardware Trace Analyzer

Keil MDK supports the Arm CoreSight® components, including the Instrumentation Trace Macrocell (ITM) and Data Watchpoint and Trace Unit (DWT). This user's guide concentrates on enabling the SWO trace in the µVision MDK. A more complete description of the MDK trace features can be found in the [Arm® Keil® MDK 5 Getting Started Guide](#). Be aware that not all debug probes support the same features. To enable SWO, click on the options for the target and select the debug tab, then select Settings (see [Figure 48](#)).

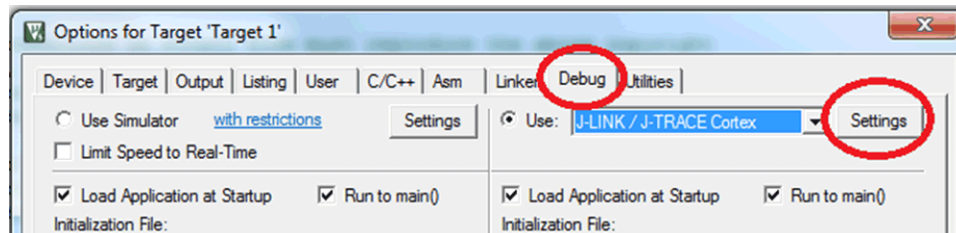


Figure 48. Debug Settings

In the Debug tab, select "SW" as port. In the Trace tab, set the flag for "Trace Enable", and make sure that the core clock is set to the correct value (see [Figure 49](#)).

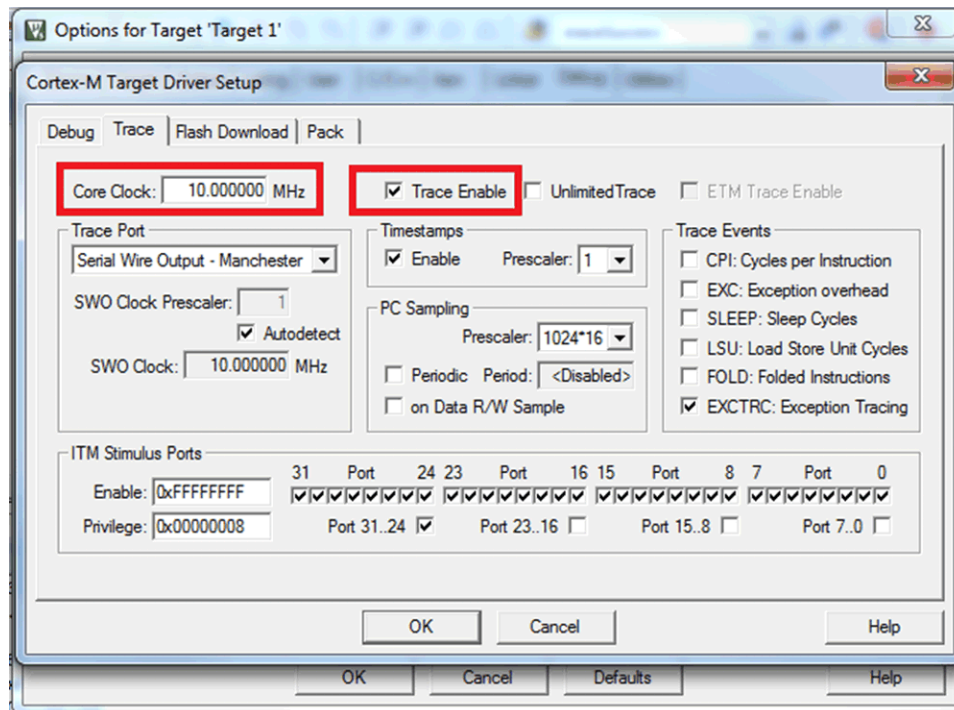


Figure 49. Enabling Trace in Project Options

Now different trace options are available:

- Exception Trace
- Event Viewer
- Logic analyzer for tracing variables over time
- Debug (printf) viewer
- Event counters

For details on the μ Vision trace features, see the [Arm® Keil® MDK 5 Getting Started Guide](#). Figure 50 shows an example exception trace for an MSP432 MCUs.

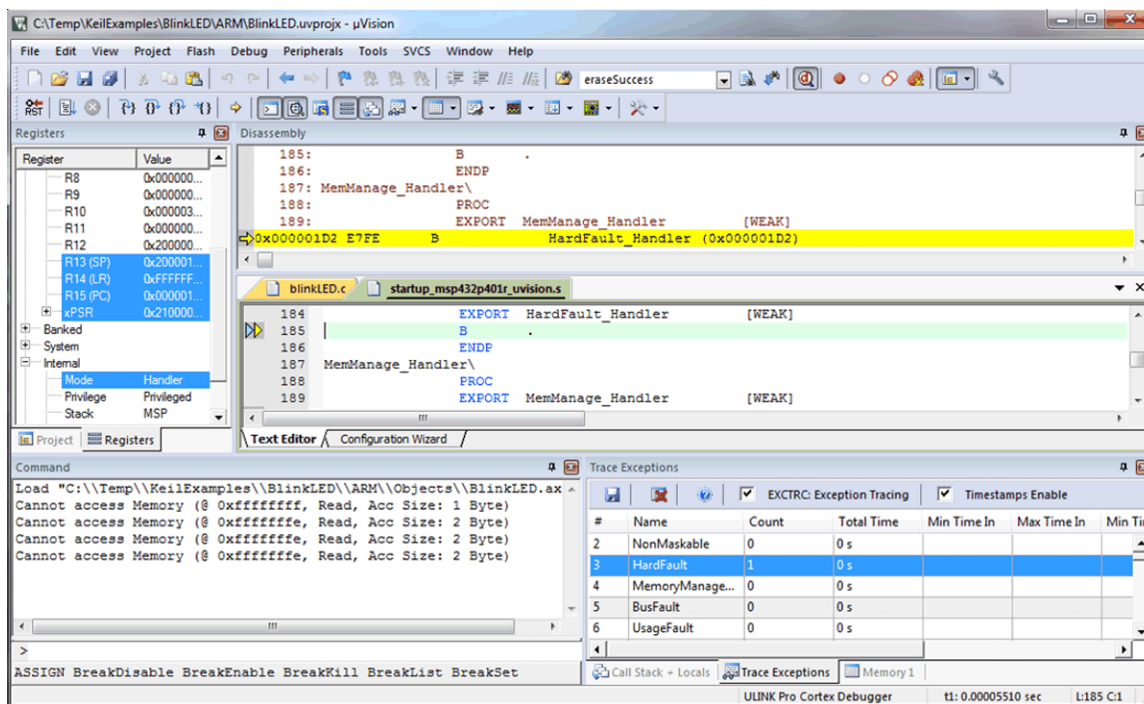


Figure 50. Exception Trace

6.6 Using ETM Trace (MSP432E4 Devices Only)

The Embedded Trace Macrocell (ETM) on some TI devices in combination with a ETM trace enabled probe like uLink Pro and Segger J-Trace can perform nonintrusive tracing of every instruction in the MCU.

To activate ETM, open the settings of the debug probe (see Figure 48). In the Cortex-M target driver setup, select the "Trace" tab and set the checkmark for "Trace Enable". In the Trace Port pulldown menu, select "Sync Trace-Port with 4 bit data", and set the check mark for "ETM Trace Enable" (see Figure 51).

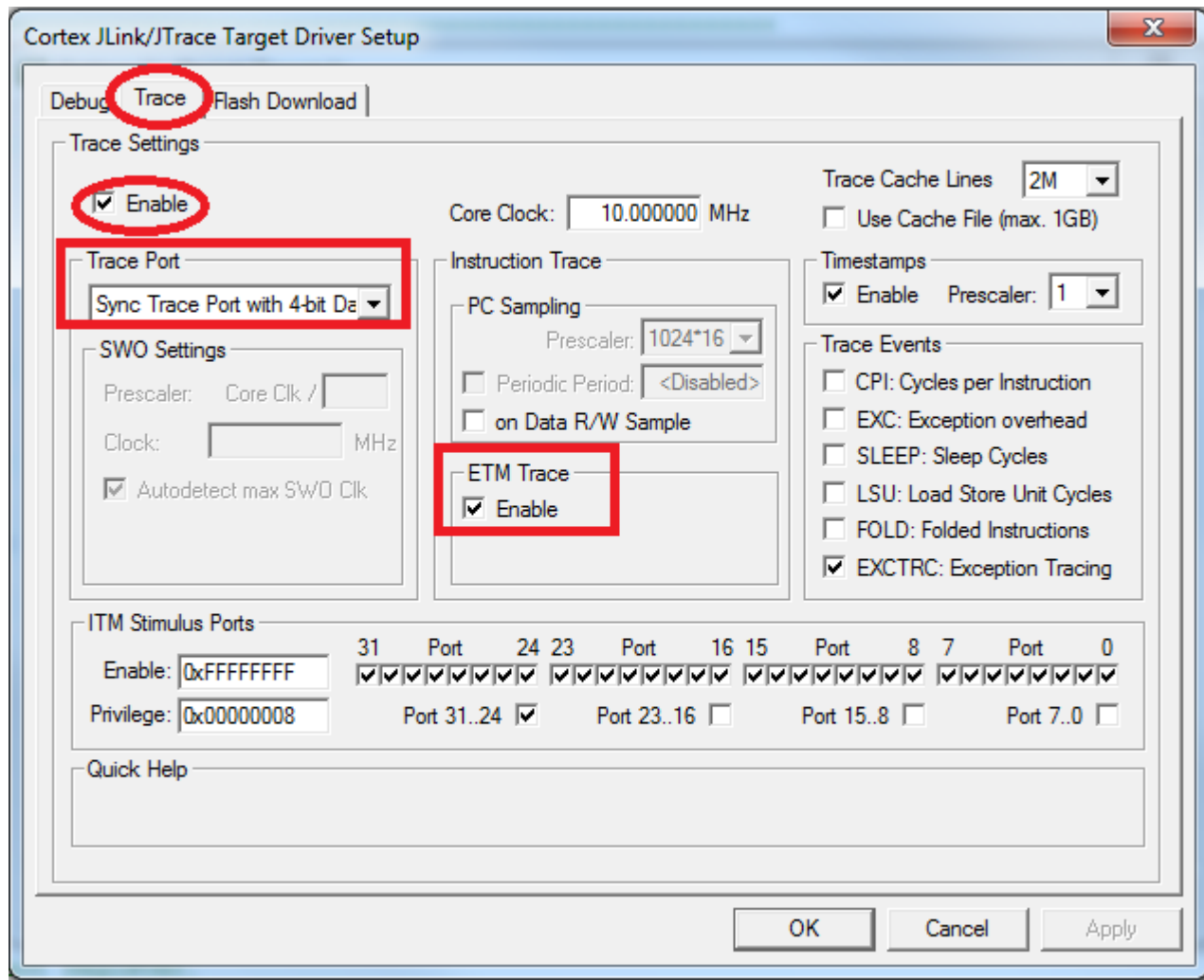


Figure 51. ETM Debug Probe Setup

Adjust any other settings as needed and go back to the project options dialog debug tab. ETM trace needs an initialization file which is provided in the CMSIS DFP examples. Add Trace.ini to the project (see [Figure 52](#)). The Trace.ini file is located in the top level of each DFP example. Should you have created a project from scratch, copy the file to the project first.

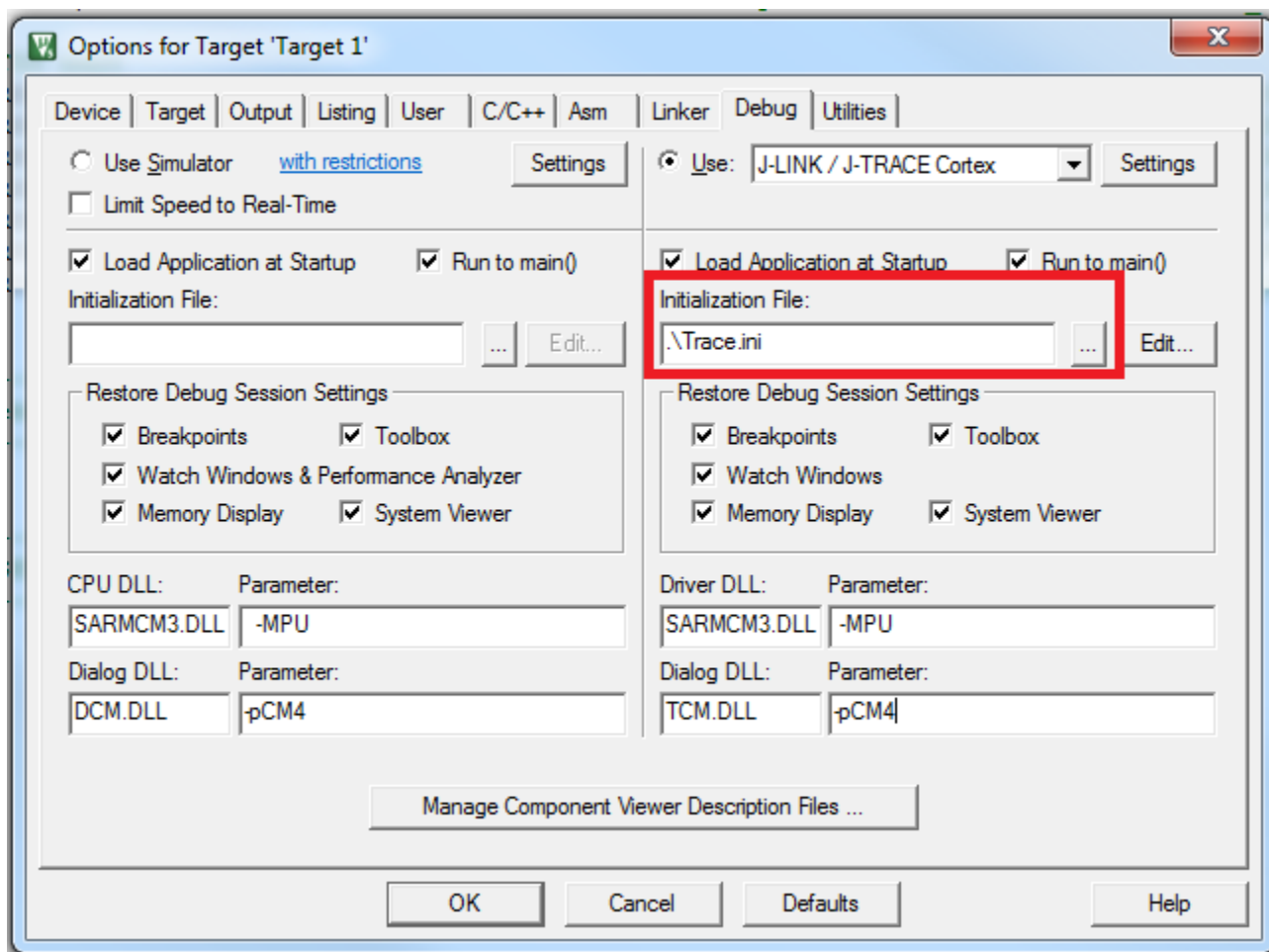


Figure 52. Adding Trace Initialization File

Trace data width is pre configured to 4 bit and ETM trace should now work out of the box. Start a debug session and open the Trace Data / Instruction Window from the View dialog (see [Figure 53](#)).



7 Frequently Asked Questions

Q: I cannot program my LaunchPad development kit; the IDE cannot connect to target. What's wrong?

A: Check the following:

- Check the debugger settings and change to Serial Wire Debug (SWD) without SWO. When the settings of Port J (PJSEL0 and PJSEL1 bits) are changed, full JTAG access is prevented on these pins. Changing to use SWD allows access through the dedicated debug pins only.

Figure 54 shows how to configure the debugger to use SWD instead of JTAG by opening the debugger settings window.

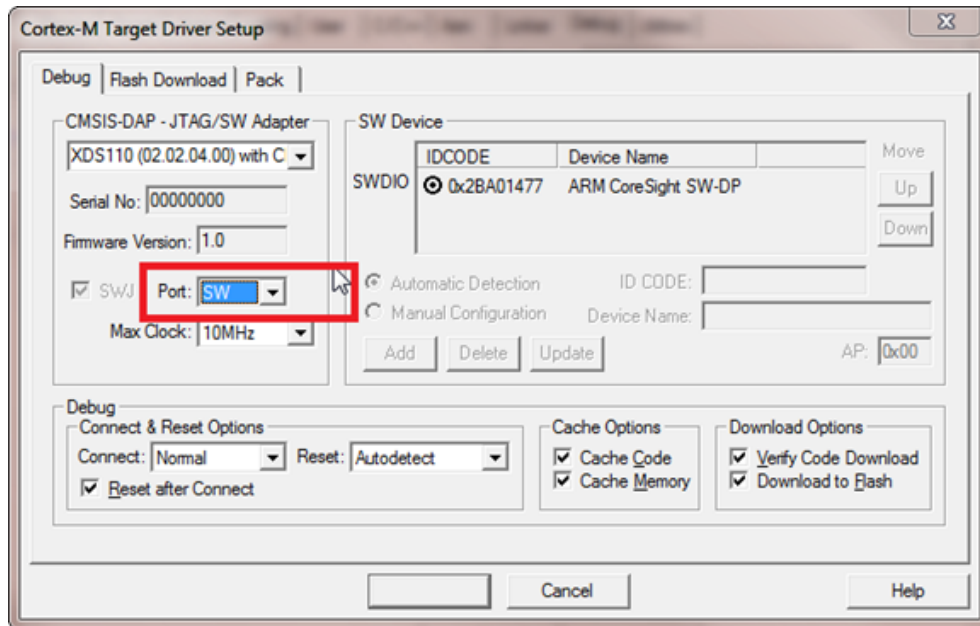


Figure 54. Changing Debugger Setting to SWD

- If even this cannot connect, reset the device to factory settings. Review the *Device Security* section of the [Code Composer Studio™ IDE 7.1+ for SimpleLink™ MSP432™ Microcontrollers User's Guide](#) for information on how to perform a factory reset on the device.

Q: Why doesn't the back-channel UART on the MSP432 LaunchPad kit work with my serial terminal program at speeds faster than 56000 baud?

A: Certain serial terminal programs such as HTerm or the CCS built-in terminal might not work with the MSP432 LaunchPad kit at specific baud rates, resulting in the software not being able to open the virtual COM port or in the baud rate getting configured incorrectly. An issue with the LaunchPad kit's emulator firmware has been identified and will be fixed in the next release. Until the update is available, use Tera Term, ClearConnex, or HyperTerminal instead or reduce the baud rate to speeds of 38400 baud or lower.

Q: Problems plugging the MSP432 LaunchPad kit into a USB 3.0 Port

A: It has been observed that when the MSP432 LaunchPad kit is connected to USB 3.0 ports provided by a certain combination of USB 3.0 host controller hardware and associated device drivers that the IDE is unable to establish a debug session with the LaunchPad kit, resulting in an error message like "CS_DAP_0: Error connecting to the target: (Error -260 @ 0x0) An attempt to connect to the XDS110 failed." in the case of Code Composer Studio. In this case the CCS-provided low-level command line utility 'xdsdfu' will also not be able to establish a connection with the LaunchPad kit.

Specifically, this issue was observed on PCs running Windows 7 that show the "Renesas Electronics USB 3.0 Host Controller" and the associated "Renesas Electronics USB 3.0 Root Hub" in the device manager. After updating the associated Windows USB drivers to more recent versions obtained from the hardware vendor the issue went away. There might be other USB 3.0 hardware and device driver combinations that will lead to the same issue. If you think you might be affected try contacting the PC vendor or try locating and installing more recent versions of the USB 3.0 device drivers. Alternatively, connect the LaunchPad kit to an USB2.0 port on the PC if available.

Q: I cannot get the backchannel UART to connect. What's wrong?

A: Check the following:

- Do the baud rate in the host's terminal application and the eUSCI settings match?
- Are the appropriate jumpers in place on the isolation jumper block?
- Probe on RXD and send data from the host. If you don't see data, it might be a problem on the host side.
- Probe on TXD while sending data from the MSP432 MCU. If you don't see data, it might be a configuration problem with the eUSCI module.
- Consider the use of the hardware flow control lines (especially for higher baud rates).

Q: After updating the MSP432 device family pack, an existing project is not compiling.

A: Try to update the device configuration file (startup_msp432.c) by right clicking on it in the project folder and selecting "Update Config File".

Q: My MSP432E device has been locked. What can I do?

A: MSP432P4xx and MSP432E devices behave differently when locked, and the unlock process also differs. See the corresponding sections in the device-specific technical reference manual or data sheet.

For MSP432E devices, The XDS debug probe supports a command line option to remove the JTAG lock. In contrast to MSP432P4xx devices, this is not assisted by the IDE.

```
C:\ti\ccs_base\common\uscif>dbgjtag.exe -f @<XDS debug probe> -Y unlock, mode=msp432e4
```

The <XDS debug probe> option in this command must be changed to specify the XDS probe in use, for example, *xds110*.

Q: Watch, Memory, and Serial Wire Viewer windows update only when the program is stopped. What can I do?

A: Set the check mark for *Periodic Window Update* in the View dialog after starting a debug session. This setting enables continuous updates.

8 Additional Keil MDK-Arm Information

For more information about Keil MDK-Arm, see the following links:

- [Keil Product Support](#)
- [Keil Microcontroller Development Kit Getting Started Guide](#)
- [Keil Microcontroller Development Kit Version 5](#)
- [Keil Microcontroller Development Kit Software Packs](#)
- [CMSIS-Pack Documentation](#)

9 References

1. [SimpleLink MSP432 SDK](#)
2. [Code Composer Studio™ IDE 7.1+ for SimpleLink™ MSP432™ Microcontrollers User's Guide](#)
3. [Texas Instruments XDS Emulation Software](#)
4. [Arm Keil MDK Getting Started Guide](#)
5. [Uniflash Standalone Flash Tool](#) with additional features for TI MCUs and TI debug probes

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from October 26, 2017 to December 12, 2017	Page
• Added "Q: My MSP432E device has been locked. What can I do?" in Section 7, Frequently Asked Questions	42
• Added "Q: Watch, Memory, and Serial Wire Viewer windows update only when the program is stopped. What can I do?" in Section 7, Frequently Asked Questions	42
• Added Uniflash (item 5) in Section 9, References	42

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2017, Texas Instruments Incorporated